Identification of Dynamics, Parameters and Synaptic Inputs of a Single Neuron using Bayesian Approach

Milad Lankarany

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

October 2013

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Milad Lankarany**

Entitled: **Identification of Dynamics, Parameters and Synaptic Inputs of a Single Neuron using Bayesian Approach**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Electrical & Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Gosta Grahne

_____ External Examiner
Dr. Pan Agathoklis

_____ External to Program
Dr. Chu Yi Su

_____ Examiner
Dr. S. Hashtrudi Zad

_____ Examiner
Dr. M. O. Ahmad

_____ Thesis Co-Supervisor
Dr. M. N. S. Swamy

_____ Thesis Co-Supervisor
Dr. Wei-Ping Zhu

Approved by _____ Dr. A. R. Sebak, Graduate Program Director _____

October 2013

Dr. C. Trueman, Interim Dean
Faculty of Engineering and Computer Science

ABSTRACT

Identification of Dynamics, Parameters and Synaptic Inputs of a Single Neuron using Bayesian Approach

Milad Lankarany, Ph.D.
Concordia University, 2013

Revealing dynamical mechanisms of the brain in order to understand how it works and processes information has recently stimulated enormous interest in computational neuroscience. Understanding the behavior of a single neuron, the most important building block of the brain, is of core interest in the brain-related sciences. Application of the advanced statistical signal processing methods, e.g., Bayesian methods, in assessing the hidden dynamics and estimating the unknown parameters of a single neuron has been considered recently as of special interest in neuroscience. This thesis attempts to develop robust and efficient computational techniques based on Bayesian signal processing methods to elucidate the hidden dynamics and estimate the unknown parameters of a single neuron.

In the first part of the thesis, Kalman filtering (KF)-based algorithms are derived for the Hodgkin-Huxley (HH) neuronal model, the most detailed biophysical neuronal model, to identify the hidden dynamics and estimate the intrinsic parameters of a single neuron from a single trace of the recorded membrane potential. The unscented KF (UKF) has already been applied to track the dynamics of the HH neuronal model in the literature. We extend the existing KF technique for the HH neuronal model to another version, namely, extended Kalman filtering (EKF). Two estimation strategies of the KF, dual and joint estimation strategies, are employed in conjunction with the EKF and UKF for

simultaneously tracking the hidden dynamics and estimating the unknown parameters of a single neuron, leading to four KF algorithms, namely, joint UKF (JUKF), dual UKF (DUKF), joint EKF (JEKF) and dual EKF (DEKF).

In the second part of this thesis, the problem of inferring excitatory and inhibitory synaptic inputs that govern activity of neurons and process information in the brain is investigated. The importance of trial-to-trial variations of synaptic inputs has recently been investigated in neuroscience. Such variations are ignored in the most conventional techniques because they are removed when trials are averaged during linear regression techniques. Here, we propose a novel recursive algorithm based on Gaussian mixture Kalman filtering for estimating time-varying excitatory and inhibitory synaptic inputs from single trials of noisy membrane potential. Unlike other recent algorithms, our algorithm does not assume an *a priori* distribution from which the synaptic inputs are generated. Instead, the algorithm recursively estimates such a distribution by fitting a Gaussian mixture model. Moreover, a special case of the GMKF when there is only one mixand, the standard KF, is studied for the same problem.

Finally, in the third part of the thesis, inferring the synaptic input of a spiking neuron as well as estimating its dynamics and parameters is considered. The synaptic input underlying a spiking neuron can effectively elucidate the information processing mechanism of a neuron. The concept of blind deconvolution is applied to Hodgkin-Huxley (HH) neuronal model, for the first time in this thesis, to address the problem of reconstructing the hidden dynamics and synaptic input of a single neuron as well as estimating its intrinsic parameters only from a single trace of noisy membrane potential. The blind deconvolution is accomplished via a novel recursive algorithm based on

extended Kalman filtering (EKF). EKF is then followed by the expectation-maximization (EM) algorithm which estimates the statistical parameters of the HH neuronal model.

Extensive experiments are performed throughout the thesis to demonstrate the accuracy, effectiveness and usefulness of the proposed algorithms in our investigation. The performance of the proposed algorithms is compared with that of the most recent techniques in the literature. The promising results of the proposed algorithms confirm their robustness and efficiency, and suggest that they can be effectively applied to the challenging problems in neuroscience.

Dedicated to my lovely family members.

Acknowledgements

I would like to sincerely appreciate both of my supervisors, Prof. M. N. S. Swamy and Prof. Wei-Ping Zhu for their continuous support, encouragement, and guidance while accomplishing this work. It has been a great honor for me to have Dr. M. N. S. Swamy and Dr. Wei-Ping Zhu as my supervisors.

I owe my deepest gratitude to Dr. M. N. S. Swamy and Dr. Wei-Ping Zhu for providing me the opportunity to work on an interdisciplinary research area ranging from neuroscience to signal processing, for effectively assisting me to conduct high impact research, and for their suggestions, ideas and invaluable insights that inspired me to do my best efforts in accomplishing this work.

I also want to express my special appreciation to Dr. Taro Toyoizumi from RIKEN Brain Science Institute for kindling my interest in computational neuroscience and sharing his vast knowledge and experience with me. Chapter 4 of this thesis has been developed under his great supervision.

I wish to thank Dr. M. Omair Ahmad and Dr. Shahin Hashtrudi Zad for their valuable comments and suggestions on my doctoral proposal and seminar, which have significantly helped shape this thesis.

Last but not the least, I am grateful to my dear Mehrnaz and all my family members, especially my parents. They were always a great source of energy that continuously motivated me during my past and current studies. Without their support, patience and constant encouragement I would not have made my studies at this point.

Contents

APPENDICES

List of Figures

List of Tables

List of Symbols

α	Gating variable due to open state of ion channels
β	Gating variable due to close state of ion channel
c_M	Membrane capacitance
ε	Observation noise
E_{Na}	Sodium reversal potential
E_K	Potassium reversal potential
E_L	Leak reversal potential
γ	Conditional probability of selecting mixands
g_E	Excitatory synaptic conductance
g_I	Inhibitory synaptic conductance
g_L	Leak conductance
g_{Na}	Sodium conductance
g_K	Potassium conductance

G	Kalman gain
I_{stim}	Stimulus current
μ_E	Mean of excitatory synaptic input
μ_I	Mean of inibitory synaptic input
N_E	Excitatory synaptic input
N_I	Inhibitory synaptic input
P	State covariance matrix
r	Parameter uncertainty
Σ_r	Covariance matrix of parameter uncertainty
σ^2_ε	Covariance matrix of observation noise
Σ_v	Covariance matrix of dynamic noise
Σ_E	Excitatory time varying variance
Σ_I	Inhibitory time varying variance
τ_E	Excitatory time constant
τ_I	Inhibitory time constant
θ	Statistical parameters

v	Dynamic noise
V	Membrane potential
$\boldsymbol{w}$	Parameter vector
$\boldsymbol{x}$	State vector
χ	Matrix of sigma points
y	Observation

List of Abbreviations

AP	Action potential
BD	Blind deconvolution
CT	Computed tomography
DE	Differential evolution
DEKF	Dual extended Kalman filtering
DUKF	Dual unscented Kalman filtering
EEG	Electroencephalography
EKF	Extended Kalman filtering
EM	Expectation maximization
EPSP	Excitatory post synaptic potential
fMRI	functional magnetic resonance imaging
FN	Fitzhugh–Nagumo model
GMKF	Gaussian mixture Kalman filtering
GMM	Gaussian mixture model
HH	Hodgkin-Huxley model

IPSP	Inhibitory post synaptic potential
JEKF	Joint extended Kalman filtering
JUKF	Joint unscented Kalman filtering
KF	Kalman filtering
LIF	leaky integrate and fire model
LNL	Linear and nonlinear model
LS	Least Square
MEG	Magnetoencephalography
ML	Maximum likelihood
NIRS	Near infrared spectroscopy
PET	Positron emission tomography
PF	Particle filtering
PSO	Particle swarm optimization
OU	Ornstein - Uhlenbeck process
UKF	Unscented Kalman filtering
VC	Voltage Clamp

Chapter 1

Introduction

1.1 General

Probably the greatest challenge of the 21th century is revealing the mechanism of the most mysterious part of the human body, the brain. Understanding human behavior − emotion, perception, movement, consciousness, etc − depends upon the identification and understanding of the behavior of the most important building block of the brain, namely, the neuron. How does it work as a single cell, communicate with other neurons and finally process the information? Understanding these neural mechanisms has a major impact on finding new therapies for a variety of mental illnesses such as, alzheimer, epilepsy and Parkinson. This is probably the main reason why some prestigious grants have been recently dedicated to projects involving mapping and simulating human brain, such as the brain activity map project (BRAIN Initiative) [1, 2] and [3], and the human brain project (HBP) [4]. In addition to the significance of the treatment of brain disorders, assessing the neural basis of human behavior has profound implications for our understanding of human learning and consciousness. As an example, advances of neural

sciences would directly influence the advances of artificial intelligence and robotics, which are of great interest in biomedical engineering.

In order to unravel the biophysical mechanisms of single neurons and consequently neural activities of the brain, various lines of research ranging from theoretical neuroscience and psychology to engineering and pure mathematics have been combined together to advance the new area, so called computational neuroscience. The term *computational neuroscience*, introduced by Eric L. Schwartz in 1985, means in general the integration of a variety of names, such as neural modeling, brain theory and neural networks. The major topics in computational neuroscience can be categorized [5] as follows: (1) single neuron modeling [6], (2) sensory processing, (3) memory and synaptic plasticity [7], (4) cognition, discrimination and learning [8], and, (5) consciousness [9].

While the brain remains one of the greatest scientific mysteries, the advent of new technologies that assist neuroscientists to open this black box has led to numerous advances in revealing the mechanisms of neurons, the major building block of the brain. Although there are several imaging techniques in the literature that demonstrate the neural activities in the brain, e.g., functional magnetic resonance imaging (fMRI), computed tomography (CT), positron emission tomography (PET), electroencephalography (EEG), magnetoencephalography (MEG), and near infrared spectroscopy (NIRS), the direct approach to understand the brain mechanisms at the level of single neurons is to probe the neuron experimentally and record the electrophysiological responses of a neuron to different (task-specific) stimuli. In this regard, there are several techniques in the literature, each of which identifies only a few dynamics underlying neural processes. The most important ones are as follows: voltage-

sensitive dye imaging [10] where the propagation of individual action potentials in the dendritic tree is observed, calcium imaging [11] that reveals the concentration of calcium in the level of single neurons, patch clamp and dynamic clamp techniques [12], [13], [14, 15] which provide the membrane potential of single neurons to be recorded, and more recently, optogenetics [16] (see also the recent amazing method [17] and references therein) which is a combination of genetics and optics to control well-defined events within specific cells of living tissue.

The main limitation of such recording techniques, e.g., the membrane potential in patch clamp technique, is that they are always uncertain due to the noise in neurons — dynamic noise — as well as errors involving the recording equipment such as the electrode resistance and capacitance-observation noise [18]. Moreover, as mentioned earlier, these techniques measure only a few dynamics of the neurons, from which it is not possible to understand the entire processes underlying the information processing of the neurons. Therefore, the major challenge today in computational neuroscience is to infer the un-observed dynamics of single neurons from such noisy and incomplete measurements. To tackle this challenge, scientists employ biophysical neuronal models that effectively represent the biological mechanisms of single neurons. Then, by using advanced computational algorithms, the parameters of these models are estimated such that the output of the model and the measured data have maximal similarities. Advanced signal processing algorithms play important roles in extracting the hidden dynamics of single neurons from noisy recorded signals. The capability of Bayesian signal processing that accomplishes the estimation process based on an *a priori* model (biophysical models for single neurons in our study), the so-called model-based technique, has been reported in

[19] and [20] to tackle several real-world problems, mostly in physics and robotics. The significance of elucidating the neural mechanisms of the brain from uncertain and incomplete electrophysiological recordings as well as their complex and highly nonlinear nature has triggered an urgent need for investigating advanced signal processing algorithms such as Bayesian approaches. As a consequence, applications of statistical signal processing algorithms such as particle filtering [21] and Kalman filtering [22] are ever increasing in computational neuroscience, particularly in the identification of the hidden dynamics of single neurons.

1.2 Literature Review

As mentioned earlier, due to the limitations of recording techniques, it is not possible to assess all the hidden dynamics and mechanisms underlying single neurons. There are several methods in the literature that employ biophysical models of single neurons to estimate unknown parameters and extract hidden dynamics of the neurons. We classify these methods into two major categories: 1) Optimization-based techniques and 2) Bayesian approaches.

Optimization-based techniques generally employ a global search algorithm, often in combination with a local search method, to minimize an objective function that measures the discrepancy between the various features of the available experimental data and the model output [21]. There are several methods in the literature, such as those in [23-31], which are used for estimating the parameters of the different neuronal models, e.g., leaky

integrate and fire (LIF), Fitzhugh–Nagumo (FN), and Hodgkin-Huxley (HH). Several optimization methods have been used for different neuronal models. The most important ones are mentioned below.

The authors of [32] used a gradient descent method to estimate the parameters of the HH model, including maximum conductances, time constants, threshold and the slope factor of activation curves. The particle swarm optimization (PSO) algorithm is employed in [26] to estimate the parameters of various spiking neuron models. A simulated annealing technique is used before running the automated fitting algorithm in [33] to reduce the exploration space. The authors of [34] compared different optimization methods for estimating the parameters of the HH model. It is shown in [34] that PSO and genetic algorithms might fail to find the correct parameters when the number of these parameters increases. More recently, the authors of [24] proposed a new variant of the differential evolution (DE) algorithm to globally estimate all the parameters of the HH model.

It is to be noted that in optimization-based techniques, the fitness function plays a critical role in the estimation process. Different fitness functions for fitting experimental data are compared in [27, 28]. A multi-objective optimization approach that combines several error functions is proposed in [31]. Genetic algorithm is used in [31] to estimate the parameters of compartmental models of neurons, given a large set of experimentally measured responses of these neurons. The excellent results of this method, in terms of estimating the parameters of the HH model and reproducing spikes, suggest that the multi-objective optimization approach [31] could serve as the building blocks for biological simulations of large neural networks.

Although optimization-based techniques try to estimate the whole set of parameters of neuronal models, yet they all need several trials of the recorded data (membrane potential, for example) to fit their parametric model. However, the authors of [30] reported that the accuracy of such estimation methods grow with the number of recorded trials. It is also noteworthy that if the trial-to-trial variations of the neural dynamics, e.g., the membrane potential, is large, the optimization-based techniques cannot estimate the parameters correctly, and might result in biased estimations [35]. Therefore, these techniques can be efficient only if the dynamical noise in the neuron is sufficiently small and the recorded data variability is ignorable over trials.

On the other hand, Bayesian approaches can efficiently estimate the parameters and track the dynamics of the neuronal models only from a single trial of recorded data. The main advantage of these methods is their capability of trial-to-trial tracking of the dynamics of neurons which, indeed, allows studying the stochastic behavior and understanding the information processing mechanism of neurons [36].

Bayesian approaches, as advanced statistical methods, have been widely used in real world applications, mostly in physics and robotics [20]. In the light of Bayesian signal processing, there are two major methods for extracting (tracking) and estimating the dynamics and intrinsic parameters of the neuronal models (especially for the HH model since it is the most detailed neuronal model) from noisy observation (e.g., membrane potential): (1) the Kalman filtering (KF) [18, 22, 36-38], and (2) the sequential Monte Carlo method (particle filtering) [21, 39]. The sequential Monte Carlo method (particle filtering [40]) is used in [39] to automatically smooth the noisy neurophysiological recordings (membrane potential) while inferring biophysically important parameters of

the HH model. In brief, the method in [39] is based on a standard expectation-maximization algorithm [41]. The first step, namely, the E-step, includes smoothing along with inferring unobserved variables of the HH model. In the second step, the M-step, new estimates of the parameters (maximum conductances) are then calculated so as to maximize the expected joint log likelihood of the observation and the inferred distribution over the hidden variables. The particle filtering algorithm associated with a robust estimation framework based on the Kitagawa's self organizing state-space model is used in [21] to simultaneously estimate the parameters and the hidden dynamics of the HH neuronal model. The authors of [21] have shown that the whole set of the HH model parameters, including time constants, reversal potentials, maximum conductances and the states (membrane voltage, sodium and potassium activation/inactivation rates) can be accurately estimated even at a high level of the observation noise with about 900 particles and a smoothing lag of 100 (samples). Kalman filtering is another Bayesian approach that has already been used for controlling the dynamics of the reduced neuronal model [42]. The authors of [18] reported, for the first time, the capability of the unscented Kalman filtering (UKF) to assimilate dynamics of spiking neuronal systems based on the HH ionic model. The feasibility of such a UKF-based framework to overcome the limitations of the dynamic clamp [12, 14, 15], a technique for creating artificial synaptic conductances, is shown in [18]. It has been pointed out in [18] that the UKF approach of predicting the hidden states of the HH ionic model can be used to control neuronal activity and pathological cellular activity such as seizures. It is shown that by adding some new dynamics to the HH model such as the extracellular potassium concentration [18, 22], the UKF can effectively track this dynamic, which plays an important role in

seizure-like neurons. Assimilating seizure dynamics has been demonstrated in [22], where the UKF has been employed to track the dynamics of two connected neurons described by the HH model. Although such frameworks are mostly verified by simulation studies, they open a new window for neuroscientists to simultaneously track the dynamics and estimate the parameters of the detailed biophysical models such as the HH model.

1.3 Motivation and Scope of the Thesis

Considering that the advanced brain-imaging techniques are not able to elucidate all dynamics and parameters of a single neuron, investigation of robust and efficient computational algorithms which reveal the intrinsic parameters and hidden dynamics of single neurons is an urgent demand in the neuroscience community. A comprehensive study on the existing methods in the literature has shown that Bayesian approaches are not only faster and more robust than optimization-based techniques, but also capable of inferring the dynamics of a single neuron from a single trial of recorded data, e.g., the membrane potential in current clamp technique. In this regard, the main objective of this thesis is to develop efficient and reliable Bayesian-based algorithms for the parameter estimation and dynamic identification of single neurons.

The first objective of the thesis is to develop the Kalman filtering (KF) algorithm for the most detailed biophysical model describing the behavior of a single neuron, the Hodgkin-Huxley (HH) neuronal model. There are two types of unknowns that have to be estimated in the HH model: the constant intrinsic parameters of the neuron and the time varying

neuronal dynamics. In this thesis, both unscented and extended Kalman filtering algorithms for identifying the dynamics of the HH neuronal model are developed based on two estimation strategies, namely, dual and joint, to address this objective.

The second objective of the thesis is inferring the excitatory and inhibitory synaptic inputs of a single neuron from a single trial of recorded membrane potential. Many existing methods in the literature can only estimate the trial mean of the synaptic inputs. Although particle filtering is recently used to infer these synaptic inputs, the distributions of these inputs should be known as the *a priori* knowledge, which is not the case in real neurons. We propose a novel recursive algorithm based on Gaussian mixture Kalman filtering that can effectively estimate the excitatory and inhibitory synaptic inputs from single trials of the membrane potential.

The third objective of the thesis is to identify the synaptic input of a single neuron as well as its intrinsic parameters and hidden dynamics. This important task is introduced in this thesis, for the first time. To fulfill our objective, we develop a recursive algorithm based on Kalman filtering followed by the expectation-maximization technique.

1.4 Organization of the Thesis

The detailed organization of the remaining chapters of the dissertation is as follows.

Chapter 2 presents the fundamentals of neural processes, which are necessary for a good understanding of the materials contained in the thesis. As the most important neural processes, action potential and synapses are explained in detail. Furthermore, five levels

of neuronal modeling, from detailed compartmental model to simple black box model, are introduced. Finally, the most important simplified neuronal models that are widely used in the literature are described.

Chapter 3 aims to reveal unobserved dynamics of a single neuron modeled by the HH neuronal model as well as its intrinsic parameters. First, a brief introduction on the Bayesian-based approaches in the literature is given. Then, the discretized version of the HH neuronal model and its state space representation are derived. Several algorithms including joint unscented KF, dual unscented KF, joint extended KF and dual extended KF are then developed. In order to provide a fair comparison between our proposed algorithms and the existing algorithm that is based on unscented Kalman filtering, different illustrative simulations are accomplished. Moreover, to verify the capability of the proposed algorithms in real world applications such as spike timing prediction, joint unscented KF and joint extended KF are applied to real data in order to predict future behavior of a real neuron.

Chapter 4 considers the problem of inferring excitatory and inhibitory synaptic conductances from a single trace of recorded membrane potential. An introduction to the existing state-of-the-art methods that employ Bayesian approach to estimate the synaptic inputs is first given. A reasonable neuronal model that can reliably mimic the sub-threshold activities of the membrane potential is introduced next. Furthermore, a general recursive framework for estimating time-varying inputs in the nonlinear systems is heuristically explained. Then, the Gaussian mixture Kalman filtering that can effectively estimate the excitatory and inhibitory synaptic inputs is derived for the neuronal model.

The performance of the Gaussian mixture Kalman filtering is compared with that of particle filtering, the most recent technique in the literature.

Chapter 5 addresses the problem of reconstructing the synaptic input as well as estimating the intrinsic parameters and the hidden dynamics (of ion channels) of a single neuron modeled by the HH neuronal model from a single trace of noisy membrane potential. After a brief review on the related works in the literature, the problem of blind deconvolution of the HH neuronal model is introduced and formulated. Then, some necessary assumptions to ensure the feasibility of the problem solution are given. A novel recursive algorithm using the extended Kalman filtering followed by an expectation-maximization algorithm is proposed to address the above mentioned blind deconvolution problem. Finally, some numerical simulations are provided to verify the accuracy and robustness of our proposed algorithm.

Chapter 6 summarizes the study undertaken in this thesis and highlights its contributions. Some suggestions for further work based on the ideas and schemes developed in this thesis are also given.

Chapter 2

Preliminaries

Unlocking the mysteries of the brain requires profound understandings of the biological basis of the neural system. Advancements in biological and biophysical representations of the neural systems provide several valuable mathematical models mimicking the behavior of neurons. These neuronal models are the main tools for leveraging our understanding about the information processing of the neurons. Therefore, a general knowledge of the biological basis of the neural systems and the mathematical basis of the neuronal models is necessary for better understanding of the neural mechanisms underlying brain functions. Providing all such materials is beyond the scope of this thesis. However, a concise and informative introduction about the biological basis of a single neuron as well as the mathematical basis of the most common neuronal models is given here. This chapter, in fact, serves as background materials to help reading the dissertation due to its multidisciplinary feature. In this chapter, firstly, a brief review on neural processes including the generation of an action potential and synapses is provided. Secondly, a brief review of different levels of neuronal modeling is presented. Thirdly, the most common simplified neuronal models are explained.

2.1 Brief Review of Neural Processes

2.1.1 Action Potential (AP)

In this sub-section, the contribution of a single neuron on receiving, integrating and transforming information is considered from the neuroscience point of view. Neuron as the most fundamental component of the brain is a specialized cell for receiving, integrating and transmitting information. A typical neuron, as shown in Figure 2.1, comprises four major components, namely, (1) dendrite and soma (postsynaptic or input compartment) that receive the excitatory and inhibitory synaptic inputs, (2) soma and axon (integrative compartment) where the input signals are integrated (in soma) and the action potential is generated (in axon hillock), (3) axon (conductile compartment) where the action potential either is zero or has a maximum value (spike) and (4) axon terminal (pre-synaptic or output compartment) where the chemical neurotransmitters are released if an action potential occurs.

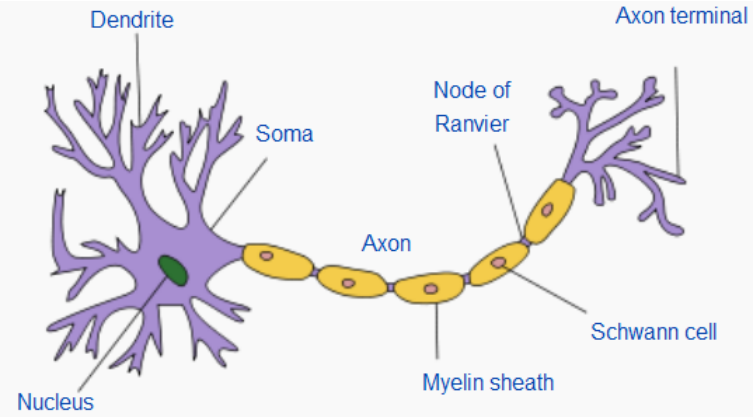


Figure 2.1. Typical shape of a single neuron and its building compartments.

The sodium (Na^+) and potassium (K^+) ions are the ones which exist most across the cellular membrane of the neuron. Using approximately 50% of a neuron's metabolic resources [43], the sodium/potassium gradient is maintained during chemical processes. In the rest state, concentration of Na^+ inside the cell is higher and therefore three Na^+ ions are transported outside the cell, while a Na^+ is entering the cell. In contrast, the higher concentration of K^+ outside the cell causes the transporting of two K^+ ions inside the cell against a K^+ outside the cell [44] (see Figure 2.2). So, the concentration gradients are followed by swapping two K^+ inside the cell for three Na^+ ions outside the cell.

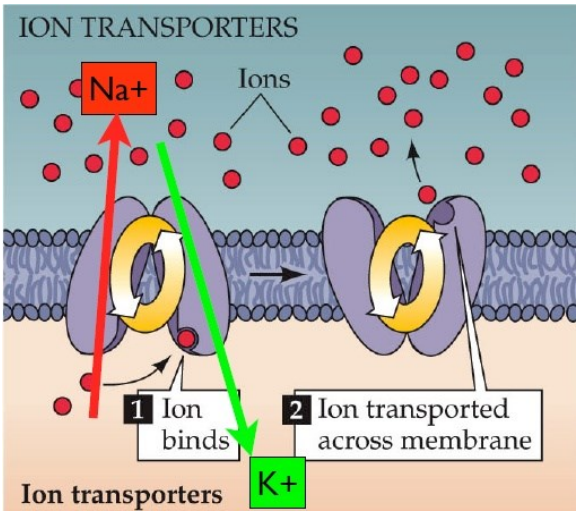


Figure 2.2. How sodium and potassium ions move across the membrane (figure from [44]).

Ions moving across the membrane are passing through the so-called ion channels. The fundamental mechanism underlying an action potential is shown in Figure 2.3. In brief, as described in [45], most of the sodium channels are closed in the rest state and the

membrane potential is determined primarily by the K^+ Nernst potential[1]. The sodium channels open if the cell is depolarized (the membrane potential increases) above some threshold. This further depolarizes the cell and permits even more sodium channels to open, allowing more sodium ions to enter the cell and forcing the cell towards the sodium Nernst potential. However, the depolarization opens potassium channels, which lead potassium ions to exit the cell. As the membrane potential moves toward the potassium equilibrium potential, the cell hyperpolarizes. The refractory period occurs when the voltage-gated[2] potassium channels close up again. During this time period, pumps exchange excess sodium ions inside the cell with excess potassium ions outside the cell [45]. This all-or-nothing process is called an action potential (AP).

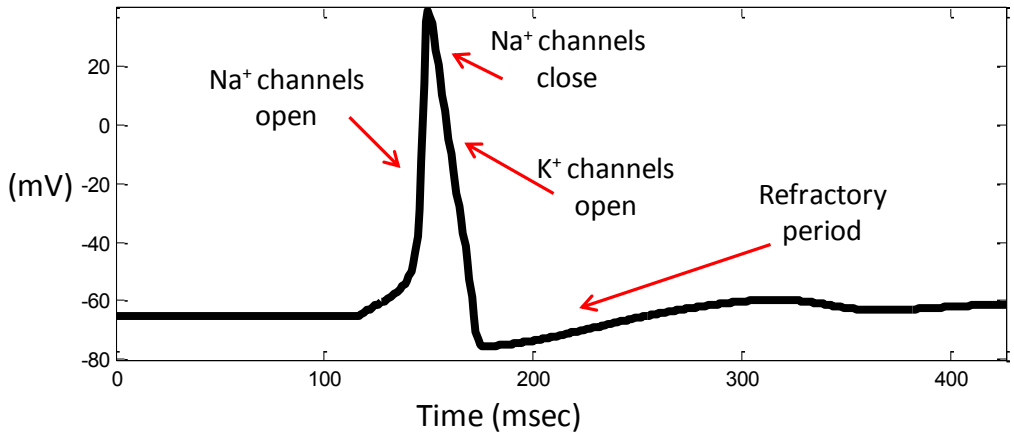


Figure 2.3. A voltage trace of a typical Action Potential (from [45]).

[1] Nernst potential for an ion is the resting potential at which the electrical and chemical driving forces cancel each other.

[2] Voltage-gated channels are referred to the ion channels whose open-close gating probabilities are functions of the membrane potential.

Moreover, when pre-synaptic neurons fire, neurotransmitters are released and diffuse across the synaptic cleft where they bind to receptors on the postsynaptic neuron (see Figure 2.4). So, the interaction between the neurotransmitters and the receptors can cause an excitatory post synaptic potential (EPSP) or an inhibitory post synaptic potential (IPSP), both of which, unlike the action potential, are passive. Usually, a single EPSP is not sufficient to trigger an action potential and multiple EPSPs are required to generate an action potential at the axon hillock[3] [46]. The IPSPs, that counter act the EPSPs, inhibit the likelihood of occurrence an AP. The interactions between the EPSPs and IPSPs can trigger a cell to fire an AP.

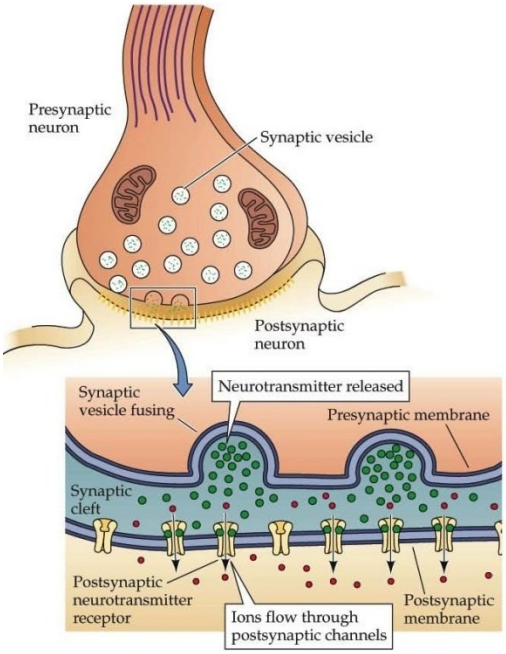


Figure 2.4. Transmission of ions across the (chemical) synapse (figure from [44]).

[3] Axon hillock is a specialized part of the neuron that is connected to the axon.

2.1.2 Synapses

There are two types of synapses in the brain, namely, chemical and electrical synapses. Neurons communicate to each other through synapses. In chemical synapses, this communication is accomplished by releasing some chemicals (message). These chemicals are referred to as neurotransmitters. In the electrical synapses, action potential of one neuron can directly fuse (pass) to another neuron and therefore no delay in information transformation occurs. It is to be noted that the electrical synapses (gap junctions) are rare in mammalian but can be found in other animals such as electric fish [46]. Figure 2.5 shows how the release of neurotransmitters through the chemical synapses causes two typical neurons to communicate.

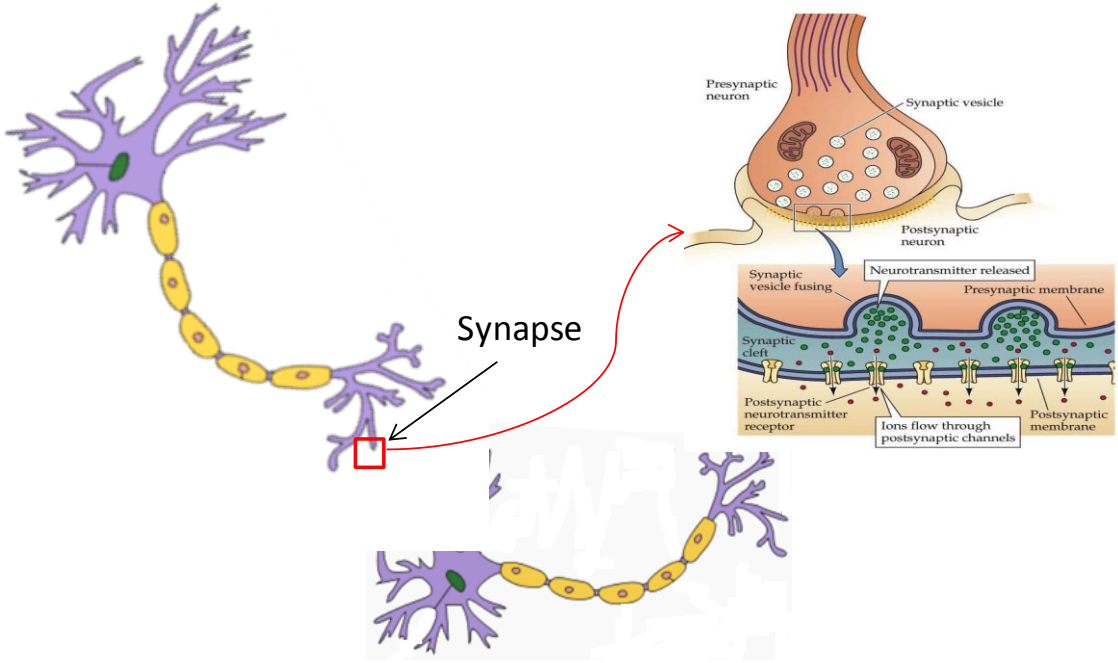


Figure 2.5. Two neurons are communicating through their chemical synapses.

The process of transmitting information across chemical synapses is accomplished in three steps, as shown in Figure 2.6. In the first step, neurotransmitters are stored in the axon terminal. In the second step, they are transported to the pre-synaptic membrane wherein they can be released in response to an action potential. Finally in the third step, the neurotransmitters activate the receptors[4] of the target cell membrane. If a neurotransmitter is able to activate the receptor of the target neuron and therefore influences its electrical excitability, it acts in only one of two ways: either to increase or to decrease the probability of producing action potentials in the target neuron. It is really interesting that despite the wide variety of synapses, they all convey messages of only two types: excitatory or inhibitory. As a result, the information that a single neuron receives is the combination of several excitatory and inhibitory messages, so called excitatory and inhibitory synaptic inputs, respectively.

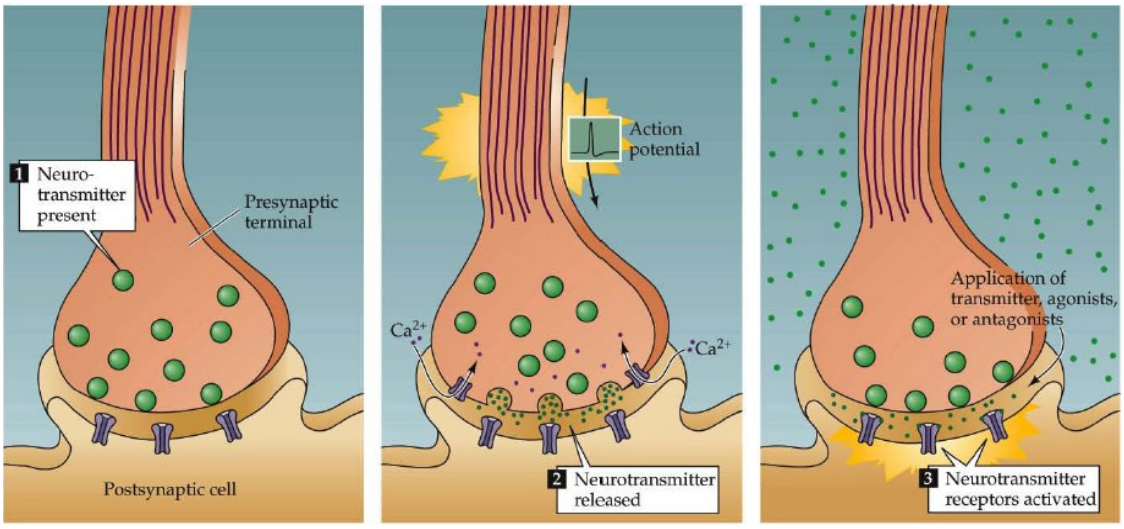


Figure 2.6. Three steps of chemical processes of neurotransmitters (figure from [44]).

[4] The receptors are ion channels selective to specific ions. When these receptors are active, they can receive the neurotransmitters released by another neuron.

2.2 Five Levels of Neuronal Modeling

A brief review of different levels of neuronal modeling is presented in this section. According to the prestigious article [47], five levels of neuronal models are classified based on their complexities. They are: Level I - detailed compartmental model, Level II - reduced compartmental model, Level III - single compartment model, Level IV - cascade model, and Level V - black box model.

2.2.1 Detailed Compartmental Model (Level I)

The cable theory of Rall [48] is used to construct the compartmental model of a single-cell neuron whose structure is described by anatomical reconstructions. In this level, the spatial structure of a neuron contributes to its dynamical behavior. Besides, as the voltage-dependent conductances are taken into account, the dynamics of a single cell are expressed by a high dimensional system of differential equations containing all spatially discretized dendrities. Such a system is nonlinear and time varying in view of the voltage difference between the membrane outside and the membrane inside the cell. One may have a better understanding about the complexity of such a system when, for example, a single neuron contains more than 1000 compartments. In fact, such a detailed model provides generating testable mechanistic hypothesis, e.g., simulation of Purkinje cells [49] or thalamocortical neurons [50]. Moreover, this level of modeling can be applied to predict the effects of extracellular electrical stimulation that is of high interest in deep brain stimulation used in the treatment of Parkinson's disease [51]. However, the detailed model suffers from a major drawback. It has high computational complexity because of the large dimensionality and intricate structures. Therefore, this level of modeling has just

limited applications, e.g., neurons with simple structures and small networks. Figure 2.7 shows how each segment (such as dendrite and axon) of a neuron is represented by an equivalent electrical circuit and consequently by differential equations.

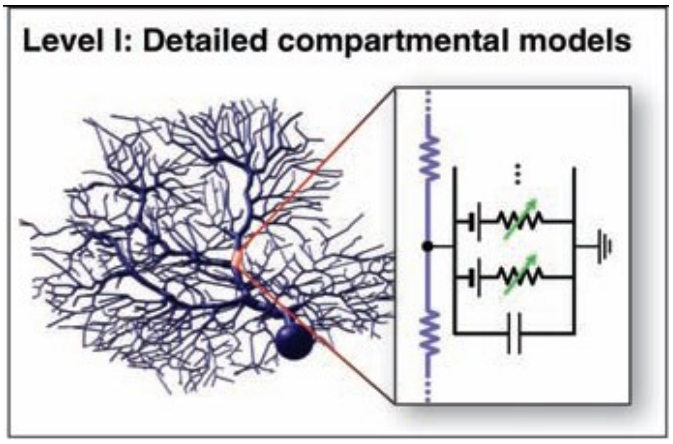


Figure 2.7. Level I modeling of single neurons (from [47]). Green arrows show the time-varying conductance.

2.2.2 Reduced Compartmental Model (Level II)

The class of reduced compartmental models with few dendritic compartments is very useful for large scale network simulations, such as the ones involving several classes of multi-compartmental cortical and thalamic neurons with more than 3000 cells [52]. This level of modeling not only can overcome the drawbacks of Level I model, but also can help to understand the somatodendritic interactions governing spiking or bursting [53]. As an example, a two-compartmental model (see Figure 2.8) is used in [54] to show that the homeostatic plasticity can follow cellular learning rules which recalibrate dendritic channels densities to yield optimal spike encoding of synaptic inputs. In conclusion,

Level II neuronal modeling is a fair compromise between realism and computational efficiency [47].

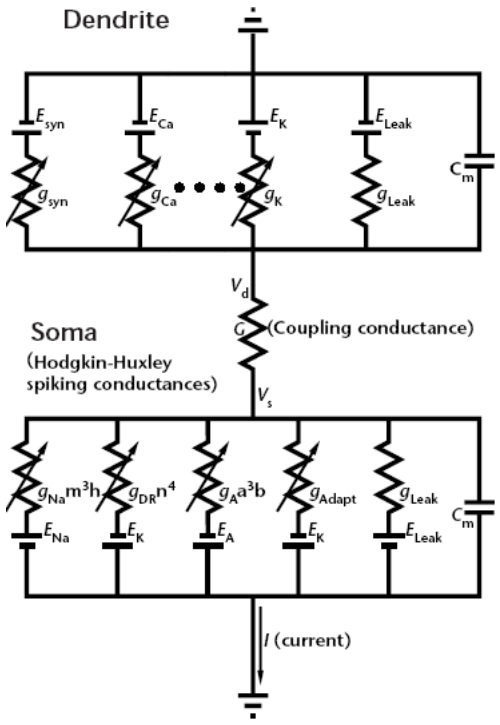


Figure 2.8. Example of Level II modeling including two compartments, each modeled by Hodgkin-Huxley formalism (from [54]). Different indices of *g* and *E* stand for different ion channels exist in the dendrite and the soma of a typical single neuron.

2.2.3 Single Compartment Model (Level III)

It has been more than 50 years since Hodgkin and Huxley proposed their model, named HH model, for generating the action potentials [55]. Although in this level of modeling the spatial structure of the neuron is not taken into account, the neuron's various ionic currents which contribute to sub-threshold behavior and spike generations are considered precisely [47]. This level of modeling provides a good quantitative understanding of

many dynamical concepts including bursting, phasic spiking and spike-frequency adaptation [56]. Another advantage of the HH model is its potential to be generalized to explain more complex dynamics, e.g., calcium currents. Figure 2.9 shows a general electrical circuit representing the HH model that comprises sodium, potassium and leak currents.

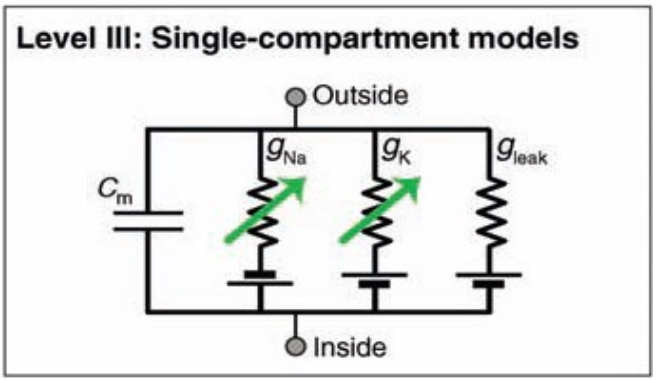


Figure 2.9. General HH model showing Level III modeling of neurons (from [47])

2.2.4 Cascade Model (Level IV)

This level is a bridge between the system neuroscience and the neural network theory, and is based on the cascade structure of a linear filter and a non-linear operator. In the studies of sensory systems for the receptive field of a neuron and the transformation of its internal activation state into a firing rate, a model that is represented by convolving the time varying input with a linear filter and applying a rectifying non-linearity is normally used [47]. This level of modeling has the advantage of simplicity and efficacy due to its linear and nonlinear (LNL) cascade structure. Therefore, this level is an appropriate model to be integrated with signal processing algorithms. As an example, the authors of

[57] showed that this level of modeling can easily track the experimental data by correlating the neural response and the stimulus (white noise) [57]. Furthermore, some studies [58] and [59] have shown that the LNL cascade structure can be obtained by more naturalistic stimulation. The block diagram of this level of modeling is shown in Figure 2.10.

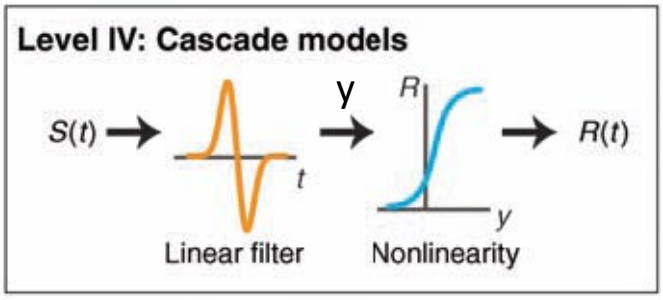


Figure 2.10. Level IV modeling, the linear non-linear cascade model (from [47]).

2.2.5 Black Box Model (Level V)

This level of modeling is completely based on signal processing algorithms and is helpful in understanding the dynamics of a neuron without considering its biophysical machinery. As discussed in [47], this modeling may indicate general principles that explain, for example, the operating points of neurons and the adaptability of neurons to alter their responses once the input statistics, such as mean and variance, are modified. Generally speaking, in this level of modeling, a neuron is considered as a black box that receives a set of time dependent inputs, sensory stimuli or spike trains from other neurons, and responds with an output spike train in which the input-output relations are characterized by a probability distribution $p(R|S)$; the probability that R (response) occurs

when *S* (stimulus) is present. Figure 2.11 shows a general block diagram of this modeling. An example of this level of modeling in neuroscience theory is the efficient coding hypothesis in visual systems [60], i.e., neural systems are adapted to the statistical structure of their sensory inputs and encode such inputs optimally. Therefore, the principal mechanism behind computation in the visual cortex can be revealed by characterizing the statistics of the sensory environment.

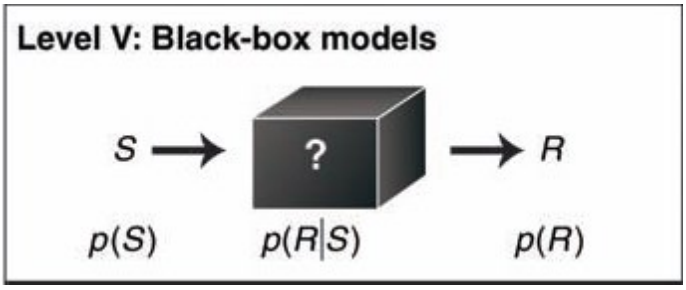


Figure 2.11. Level V modeling of single neuron (from [47]).

In fact, basic computations, e.g., addition, subtraction, multiplication, as well as task-specific computations, e.g., motion detection, and sound localization, which are accomplished by single neurons assess the information processing of single neurons. Several studies in the literature (see [47] for more detail) have shown the application of the aforementioned five levels of modeling to study such information processing of single neurons.

2.3 Simplified Neuronal Models

In this section, the most well-known simplified neuronal models, which mostly belong to the third and fourth levels of modeling (see Section 2.2), are briefly introduced.

2.3.1 Leaky Integrate and Fire (LIF) Model

The LIF model, proposed by Louis Lapicque in 1907, is probably the best known example of a formal spiking neuronal model. This model, in its basic representation, is a RC circuit that is charged by an external current. The LIF can be expressed as,

$$C\frac{dV}{dt}+\frac{V}{R}=I \tag{2.1}$$

where C is the capacitance representing the capability of the cell's membrane to store electrical charges (ions), R is the leak resistance and V is the membrane potential, i.e., the difference between the voltage inside and the voltage outside the cell membrane. In LIF model, as shown in Figure 2.12, when the steady state solution of (2.1) is greater than a threshold, V is set to a maximal value that indicates the firing of a neuron. The electrical circuit representing the LIF behavior is shown in Figure 2.12.

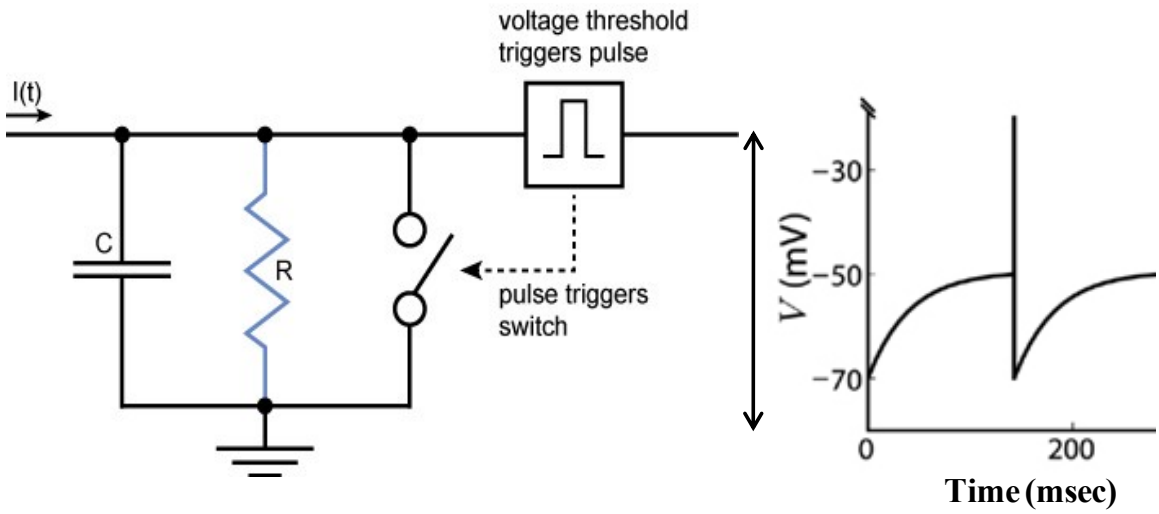


Figure 2.12. An electrical circuit representing the leaky integrate and fire model.

Although this model seems simple, it can nicely mimic the sub-threshold activities of single neurons. It is worthy to note that this property of LIF makes it particularly useful for analyzing synaptic activities of neurons where the active channels (spiking) are pharmacologically blocked. This is the main reason that we are employing this neuronal model for inferring the excitatory and inhibitory synaptic conductances (see Chapter 4). Moreover, stronger and biologically more relevant extensions of LIF are the exponential integrate and fire model as well as adaptive exponential integrate and fire model (AdEx) [61].

2.3.2 Fitzhugh–Nagumo (FN) Model

The FN model was suggested by Richard Fitzhugh in 1961 and the equivalent circuit was created by J. Nagumo [62]. This model comprises two differential equations which describe a prototype of an excitable system (e.g., a neuron). The FN model is expressed as follows.

$$\begin{aligned} \frac{dV}{dt} &= f(V) - w + I_{ext} \\ \frac{dw}{dt} &= a(bV - cw) \end{aligned} \tag{2.2}$$

where V is the membrane potential that allows regenerative self-excitation (via a positive feedback), w is the recovery variable that provides a slower negative feedback, $f(V)$ is a polynomial of third degree, a, b, and c are constant parameters, and I_{ext} is the external stimulus to be injected to the neuron. If the amplitude of this stimulus exceeds a certain value (threshold), the membrane potential fires. In computational neuroscience, the FN

model is considered as the two dimensional simplification of the HH model and is widely used in the literature ([63] and see [62] and references therein). The FN model has recently been modified using time-varying spiking threshold [63] which can compensate some of the limitations of this model.

2.3.3 Morris-Lecar Model

This model is introduced by Morris and Lecar in 1981 [64]. It can be interpreted as an extension of FN model to include voltage gated calcium channel with a delayed rectifier potassium channel [64]. This model is described as follows.

$$\begin{aligned} C\frac{dV}{dt} &= -g_{Ca}M_{\infty}(V-E_{Ca}) - g_K w(V-E_K) - g_L(V-E_L) + I_{ext} \\ \frac{dw}{dt} &= \frac{(W_{\infty}-w)}{T_{\infty}} \end{aligned} \tag{2.3}$$

where the definitions of V, w and I_{ext} are the same as in the FN model, g_K, g_{Ca} and g_L are the conductances of potassium, calcium and leak currents, E_K, E_{Ca} and E_L are the corresponding reverse potentials, and $T_{\infty}(V)$ is the time constant for the potassium channel relaxation in response to the changes of voltage which can be given as follows.

$$T_{\infty}(v) = T_0 \sec h[(V-V_3)/2V_4] \tag{2.4}$$

where V_3 and V_4 are constant parameters, and T_0 is the time scale for the recovery process. The open state probability functions, M_{∞} and W_{∞}, are given as:

$$\begin{aligned} M_{\infty}(V) &= (1+\tanh[(V-V_1)/V_2])/2 \\ W_{\infty}(V) &= (1+\tanh[(V-V_3)/V_4])/2 \end{aligned} \tag{2.5}$$

where V_1 and V_2 are constant parameters. It is worthy to note that all of the parameters in the Morris-Lecar equations are experimentally measurable. Thus, this simple model can effectively mimic a wide range of phenomena, e.g., fast spiking and bursting, that occur in different excitable systems (see [64] for more details).

2.3.4 Hodgkin-Huxley (HH) Model

The Hodgkin–Huxley (HH) model, introduced by A. L. Hodgkin and A. F. Huxley in 1952, describes how action potentials in neurons are initiated and propagated. It uses a set of nonlinear ordinary differential equations to approximate the electrical characteristics and dynamics of a single neuron. To provide a brief description of this model, consider the equivalent lumped circuit representing the HH model in Figure 2.13. As can be seen from this figure, the HH neuronal model contains three components [45]: (1) conductors (or resistors) representing the ion channels; (2) batteries representing the concentration gradients of the ions; and (3) capacitors representing the ability of the membrane to store the charge (see chapter 1 of [45] for more details). Each ion channel (K^+ and Na^+) in the HH model is characterized by a time varying conductor (g_K and g_{Na}) in series with a battery (E_K and E_{Na}) and the leak current is modeled by a constant leak conductor (g_L) in series with a battery (E_L).

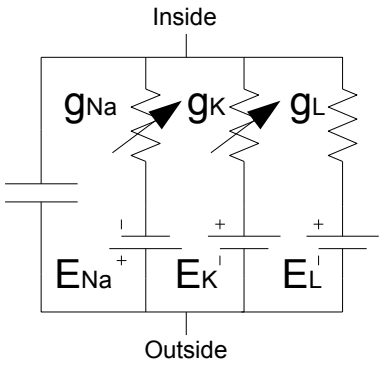


Figure 2.13. Schematic representation of the HH neuronal model

The HH neuronal model is the most detailed biophysical model to describe dynamical behaviors of a single neuron. In this model, the transmembrane potential, $V = V_{\text{inside}} - V_{\text{outside}}$, of a single neuron can be described by the following differential equations [45]:

$$\begin{aligned} C_M \frac{dV}{dt} &= -g_{Na} m^3 h (V - E_{Na}) - g_K n^4 (V - E_K) - g_L (V - E_L) \\ \frac{dn}{dt} &= \alpha_n(V)(1-n) - \beta_n(V) n \\ \frac{dm}{dt} &= \alpha_m(V)(1-m) - \beta_m(V) m \\ \frac{dh}{dt} &= \alpha_h(V)(1-h) - \beta_h(V) h \end{aligned} \tag{2.6}$$

where C_M is the membrane capacitance, and m, n and h are the gating variables considered as the dynamics of the HH model. The equations describing these dynamics are given below [45].

$$\begin{aligned} \alpha_n(V) &= 0.01(V+55)/(1-\exp(-(V+55)/10)) \\ \beta_n(V) &= 0.125\exp(-(V+65)/80) \\ \alpha_m(V) &= 0.1(V+40)/(1-\exp(-(V+40)/10)) \\ \beta_m(V) &= 4\exp(-(V+65)/18) \\ \alpha_h(V) &= 0.07\exp(-(V+65)/20) \\ \beta_h(V) &= 1/(1+\exp(-(V+35)/10)) \end{aligned} \tag{2.7}$$

It is clear from (2.7) that α_q and β_q (for $q = n, m, h$) are nonlinear functions of the membrane potential, V. It is to be noted that though we have used the known parameters for α_q and β_q as used in [18, 37, 39, 45, 65], they can be expressed in a general form. Let $q = n, m, h$ be the gating variable represented by a first order differential equation as follows.

$$dq/dt = \frac{q_\infty(V) - q}{\tau_q(V)} \tag{2.8}$$

where $q_\infty(V)$ and $\tau_q(V)$ are the steady state and the time constants of the gating variable q, respectively. The function $q_\infty(V)$ is a sigmoid function of the membrane potential expressed as:

$$q_\infty(V) = \frac{1}{1 + e^{-s_q(V - V_{th(q)})}} \tag{2.9}$$

where s_q and $V_{th(q)}$ stand for the slope and threshold of the steady state curve, respectively, and

$$\tau_q(V) = t_q \cosh^{-1} \frac{s_q(V - V_{th(q)})}{2} \tag{2.10}$$

with t_q being time constant factor of the q.

The HH neuronal model can mimic almost all the behaviors of a single neuron. This model, that is considered as the most biophysically detailed neuronal model, will be employed in Chapters 3 and 5 of the thesis.

2.4 Summary

A brief review on biological processes underlying the generation of an action potential and synapses has been provided in this chapter. Then, different levels of neuronal modeling from detailed compartmental model to black box model have been described. Finally, the most common simplified neuronal models, namely, leaky and integrate fire (LIF), Fitzhugh-Nagumo (FN), Moris-Lecar, and Hodgkin-Huxley (HH), are explained. It is to be noted that the LIF model will be used in Chapter 4 of the thesis where inferring the excitatory and inhibitory synaptic inputs is of our interest. Furthermore, the HH neuronal model will be used in Chapters 3 and 5 where parameter estimation and dynamic reconstruction of the spiking neurons are investigated.

Chapter 3

Dynamic Tracking and Parameter Estimation of Hodgkin-Huxley Neuronal Model

3.1 Introduction

Understanding the brain's mechanisms relies on assessing the dynamics of its building blocks. Identification of the dynamics of a single neuron, the most important building block of the brain, is a challenging problem in neuroscience. The main reason is that there are several hidden dynamics and parameters in the neurons that are not measurable even using recent brain-imaging techniques. A general procedure for estimating these hidden variables is stimulating a neuron with a specific input (e.g., current injection) in different trials and recording the corresponding responses (e.g., the membrane potential). To match this input-output relationship, a reasonable neuronal model that can reliably mimic the neuronal behavior should be used. Then, estimation methods are used to

calculate the unknown parameters of the model. Figure 3.1 shows a block diagram representation of a system X whose parameters should be estimated such that the system output and the recorded membrane potential from a real neuron have maximum similarities. Hodgkin-Huxley neuronal model is the best candidate since it is the most biophysically-detailed model among other neuronal models introduced in Section 2.3. Since this model is represented by highly nonlinear differential equations, revealing the dynamics and estimating the parameters of this neuronal model is extremely complex and computationally expensive.

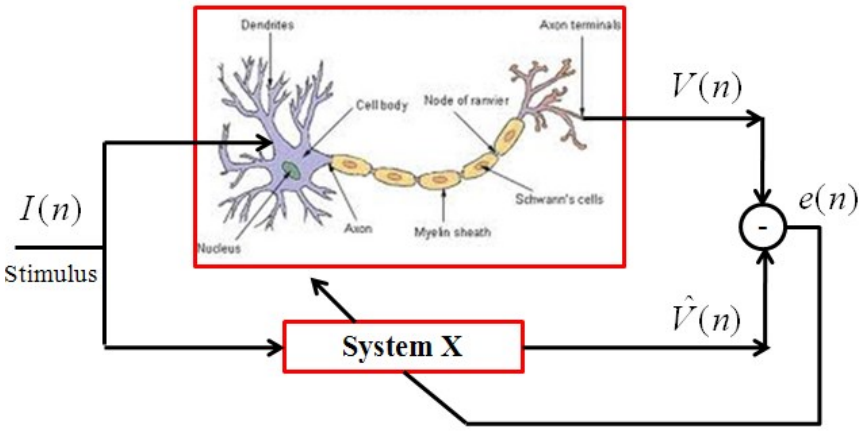


Figure 3.1. Schematic representation of modeling a real neuron. System X is designed to track the dynamics of a real neuron. System X stands for a biophysical neuronal model, e.g., HH model. V, and $\hat{V}$ express the membrane potential of the real and modeled neuron, respectively. I is the stimulus, e.g., injected current, of the neuron.

As mentioned in Section 1.2, there are two categories of the algorithms that estimate the parameters and identify the dynamics of the single neurons, namely, optimization-based and Bayesian-based methods. It has been pointed out Chapter 1 that the Bayesian-based methods in general outperform the optimization-based techniques, since the latter cannot reveal the variability of the neuronal dynamics in different trials. In the category of

Bayesian approaches, unscented Kalman filtering (UKF) [18] and particle filtering (PF) [39] and [21] have already been derived for the HH neuronal model. In accordance with our descriptions in Section 1.2, the PF-based methods need a large number of particles to perform accurate estimation. However, KF-based algorithms are generally more useful for real-time applications such as dynamic clamp [18].

We investigate, in this chapter, the feasibility of the extended KF (EKF) in extracting time-varying dynamics and unknown biophysical parameters (maximum conductances) of a single neuron described by the HH neuronal model. Since the hidden dynamics and unknown parameters of the HH neuronal model have to be calculated simultaneously, two estimation strategies, namely, dual and joint strategies [66-68] are used in conjunction with both EKF and UKF algorithms to develop four techniques, namely, joint unscented Kalman filter (JUKF), dual unscented Kalman filter (DUKF), joint extended Kalman filter (JEKF) and dual extended Kalman filter (DEKF) for the HH neuronal model. The precision of these four algorithms as well as their speed and computational cost are also studied in order to determine the most appropriate KF-based algorithm for real time applications. The accuracy of the above mentioned KF approaches are verified under different signal to observation noise ratios. Furthermore, the JUKF and JEKF algorithms are extended to estimate various parameters of entire HH neuronal model including the kinetics of the ion channels. These algorithms are applied to real data recorded from the membrane potential of a single neuron to estimate its ion channel kinetics and consequently reveal the underlying dynamics.

3.2 State Space Representation of the HH Neuronal Model

The Hodgkin–Huxley (HH) neuronal model has already been introduced in Section 2.3. To better describe the state space representation of the HH neuronal model, its governing equations are repeated below,

$$\begin{aligned} C_M \frac{dV}{dt} &= -g_{Na} m^3 h(V - E_{Na}) - g_K n^4 (V - E_K) - g_L (V - E_L) \\ \frac{dn}{dt} &= \alpha_n(V)(1-n) - \beta_n(V) n \\ \frac{dm}{dt} &= \alpha_m(V)(1-m) - \beta_m(V) m \\ \frac{dh}{dt} &= \alpha_h(V)(1-h) - \beta_h(V) h \end{aligned} \tag{3.1}$$

where C_M is the membrane capacitance, and m, n and h are given in (2.7).

By defining vector $x=[V,n,m,h]^H$, the above general HH model can be rewritten in state space domain as:

$$\dot{\boldsymbol{x}}(t) = F[\boldsymbol{x}(t)] = \begin{bmatrix} F_1(\boldsymbol{x}(t)) \\ F_2(\boldsymbol{x}(t)) \\ F_3(\boldsymbol{x}(t)) \\ F_4(\boldsymbol{x}(t)) \end{bmatrix} = \begin{bmatrix} \dfrac{-g_{Na} m^3 h(V - E_{Na}) - g_K n^4 (V - E_K) - g_L (V - E_L)}{C_M} \\ \alpha_n(V)(1-n) - \beta_n(V) n \\ \alpha_m(V)(1-m) - \beta_m(V) m \\ \alpha_h(V)(1-h) - \beta_h(V) h \end{bmatrix} \tag{3.2}$$

where $F[\boldsymbol{x}(t)]$ is the transient function. There are always sources of neuronal noise which may have originated from the membrane channel stochasticity, branch point conduction failure, and probabilistic transmitter release [18]. Therefore, as discussed in [18, 22], these uncertainties, the so called system noise, can be considered as an additive white

Gaussian noise. In view of the limitations of the imaging techniques [18], it is impossible to measure all the necessary biophysical variables describing a single neuron model. For instance, in intercellular electrophysiological recordings, the membrane potential is the only measurable variable. This measurement, on the other hand, may contain noise from the recording equipment, which is known as observation noise. Therefore, the HH model (3.1) can be expressed as a state space model that includes such uncertainties.

$$\begin{cases} \dfrac{d\boldsymbol{x}(t)}{dt} = F[\boldsymbol{x}(t)] + BI_{Inj}(t) + \boldsymbol{v}(t) \\ y(t) = C\boldsymbol{x}(t) + \varepsilon(t) \end{cases} \tag{3.3}$$

where C=[*1,0,0,0*], $\boldsymbol{v}$ and ε are the system noise and observation noise, respectively, $I_{Inj}(t)$ is the external injected current and B=$[1/C_M,0,0,0]^H$. From (3.1) and (3.2) one can easily see that $F[\boldsymbol{x}(t)]$ is highly nonlinear due to $\boldsymbol{x}(t)$. Therefore, the HH dynamical model represents a highly nonlinear and time-varying system. To estimate the dynamics ([V, n, h, m]) of the HH model, we employ the nonlinear versions of KFs, i.e., UKF and EKF. Obviously, as the membrane potential is not recorded continuously, the Euler discretization can be employed (see [18, 39]). Since the sampling frequency in electrophysiological recordings is sufficiently high (above 10 KHz), Euler discretization has been used in all the methods in the literature. However, investigation of the effect of other discretization strategies, such as zero-order hold, on the accuracy of the estimated states and parameters is an interesting topic which should be considered in future studies.

Here, for the sake of simplicity for deriving the KF for the HH model, we assume that the sampling rate is equal to unity for the rest of this chapter. In this case, (3.3) can be written as:

$$\begin{cases} \boldsymbol{x}(k) = \boldsymbol{x}(k-1) + F\left[\boldsymbol{x}(k-1)\right] + BI_{Inj}(k-1) + \boldsymbol{v}(k-1) \\ y(k) = C\boldsymbol{x}(k) + \varepsilon(k) \end{cases} \tag{3.4}$$

We will discuss the actual sampling rate in Section 3.6 and consider how the sampling rate would affect our algorithms in Section 3.7. In order to tackle the state space equation (3.4) that involves a highly nonlinear function $F[\boldsymbol{x}]$, two main nonlinear versions of the KF will be described in the next section. Please note that the mathematical symbols used in this chapter for describing KF algorithms are consistent with those in [67]. Moreover, the symbol Σ (small sigma whose indices appear as subscript and superscript) is used in the thesis to represent the covariance matrix [69].

3.3 Kalman Filtering (KF)

Before describing the nonlinear versions of the KF which are employed in this chapter, a brief introduction about the KF is given here. Generally speaking, KF [70] uses a set of mathematical equations underlying the process model to estimate the current state of a system and then correct it using any available sensor measurements [70]. As the standard Kalman filtering equations are derived for linear time invariant systems, let us assume that a system $\boldsymbol{S}$ is represented by the following state space model.

$$S: \begin{cases} x(k) = A\boldsymbol{x}(k-1) + BI_{Inj}(k-1) + \boldsymbol{v}(k-1) \\ y(k) = C\boldsymbol{x}(k) + \varepsilon(k) \end{cases} \tag{3.5}$$

where, similar to (3.4), $\boldsymbol{x}$ and y express the state vector and the observation, respectively, and A and C are the transition and observation matrices. Now, assuming that the

observation measurement *y(k)* and the state space model (3.5) are given, KF can be considered as an estimator that produces three types of outputs [19]. It can be thought of as a *state estimator or reconstructor*, i.e., it modifies the state estimation of $\boldsymbol{x}(k)$ from noisy measurements $\{y(k)\}_{0:k}$. Second, KF can be considered as a *measurement filter* that, on its input, accepts the noisy sequence $\{y(k)\}_{0:k}$ and, on its output, produces a filtered measurement sequence, $\hat{y}(k/k)$. Finally, the processor can be thought of as a *whitening filter* that accepts noisy correlated measurements $y(k)$ and produces uncorrelated or equivalent white measurements $e(k) = y(k) - \hat{y}(k/k)$, the so-called innovation sequence. KF contains two major steps for updating (modifying) the estimate of the state $\boldsymbol{x}(k)$ due to the arriving new observation $y(k)$. The first step is the *prediction step*, in which the *a priori* estimates of the state and covariance matrix, denoted as $\hat{\boldsymbol{x}}_{k|k-1}$ and $\Sigma_{\hat{\boldsymbol{x}}_{k|k-1}}$, are calculated as follows.

$$\hat{\boldsymbol{x}}_{k|k-1} = A\hat{\boldsymbol{x}}_{k-1|k-1} + BI_{Inj}(k) \tag{3.6}$$

$$\Sigma_{\hat{\boldsymbol{x}}_{k|k-1}} = A\Sigma_{\hat{\boldsymbol{x}}_{k-1|k-1}}A^H + \Sigma_{\boldsymbol{v}} \tag{3.7}$$

where $\Sigma_{\boldsymbol{v}}$ is the covariance matrix of the dynamical noise $\boldsymbol{v}$. Then, the innovation $e(k) = y(k) - \hat{y}(k/k)$ and its covariance matrix $\Sigma_{ee}(k) = C\Sigma_{\hat{\boldsymbol{x}}_{k|k-1}}C^H + \sigma^2_{\varepsilon}$ are calculated, where σ^2_{ε} is the variance of the observation noise (note that the dimension of the observation in our case is one). In the second step, i.e., the *correction step*, the *a posteriori* estimates of the state and covariance matrix, denoted as $\hat{\boldsymbol{x}}_{k|k}$ and $\Sigma_{\hat{\boldsymbol{x}}_{k|k}}$, are calculated as follows.

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + G_{\boldsymbol{x}_k}[y(k) - C\hat{\boldsymbol{x}}_{k|k-1}] \tag{3.8}$$

$$\Sigma_{\hat{\boldsymbol{x}}_{k|k}} = \left(I - G_{\boldsymbol{x}_k} C\right)\Sigma_{\hat{\boldsymbol{x}}_{k|k-1}} \tag{3.9}$$

where $G_{\boldsymbol{x}_k}$ is the so called Kalman gain and can be obtained by

$$G_{\boldsymbol{x}_k} = \Sigma_{\hat{\boldsymbol{x}}_{k|k-1}} C^H \left(C\Sigma_{\hat{\boldsymbol{x}}_{k|k-1}} C^H + \sigma_\varepsilon^2\right)^{-1} \tag{3.10}$$

3.3.1 Extended KF (EKF)

In the EKF [19, 67], a first-order Taylor linearization of the nonlinear process and measurement models are used to derive the underlying prediction-correction mechanism. Using (3.4), the discretized HH model, *a priori* (predicted) state estimate and error covariance matrix can be calculated at each time bin *k* (Note that the time index *k* takes integer values only). Moreover, following the standard KF for the LTI systems, the correction step calculates *a posteriori* state estimate and error covariance matrix for this time instant. These variables will be used in the KF framework for the next time instant *k+1*, upon the arrival of new observation. Here a general mathematical description of the EKF is given. Using (3.4), *a priori* (predicted) state estimate can be written as:

$$\hat{\boldsymbol{x}}_{k|k-1} = \hat{\boldsymbol{x}}_{k-1|k-1} + F[\hat{\boldsymbol{x}}_{k-1|k-1}] + BI_{Inj}(k) \tag{3.11}$$

The *a priori* estimate of the error covariance matrix is given by:

$$\Sigma_{\hat{\boldsymbol{x}}_{k|k-1}} = A_{\hat{\boldsymbol{x}}_{k|k-1}} \Sigma_{\hat{\boldsymbol{x}}_{k-1|k-1}} A^H_{\hat{\boldsymbol{x}}_{k|k-1}} + \Sigma_{\boldsymbol{v}} \tag{3.12}$$

where $A_{\hat{\boldsymbol{x}}_{k|k-1}} = I + \frac{\partial F}{\partial \boldsymbol{x}} |_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k-1|k-1}}$, I is an identity matrix, and $\Sigma_{\boldsymbol{v}}$ is the covariance matrix of the process noise. Following the conventional KF for the linear time invariant (LTI) systems, the correction step calculates *a posteriori* state estimate as follows:

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + G_{\boldsymbol{x}_k}[y(k) - C\hat{\boldsymbol{x}}_{k|k-1}] \tag{3.13}$$

where $G_{\boldsymbol{x}_k}$ can be obtained by (3.10). Moreover, *a posteriori* estimate of the error covariance matrix can be updated as

$$\Sigma_{\hat{\boldsymbol{x}}_{k|k}} = \left(I - G_{\boldsymbol{x}_k} C\right)\Sigma_{\hat{\boldsymbol{x}}_{k|k-1}} \tag{3.14}$$

As can be seen from (3.9) and (3.14), $\Sigma_{\hat{\boldsymbol{x}}_{k|k}}$ are the same for the standard KF and EKF. The main difference between EKF and the standard KF in our case, since the observation matrix is constant, is in calculating the transition matrix $A_{\hat{\boldsymbol{x}}_{k|k-1}}$ at each time step *k*.

3.3.2 Unscented KF (UKF)

In the UKF [71-73], *a posteriori* probability density function of the estimated state is approximated by a Gaussian distribution and the mean and covariance of the estimated system are propagated by specifying an ensemble of points, *sigma points*, that characterize statistical properties of the states. In fact, unlike the EKF that linearizes the nonlinear process function by using Jacobian matrices, the UKF uses the sigma points to capture the mean and covariance estimates.

Given the state vector at time step *k-1*, one can compute a collection of sigma points and store them in the columns of the matrix χ_{k-1} of dimension $L \times (2L+1)$, where L is the

dimension of the state vector. Please note that in the mathematical symbols used in this thesis, *k* stands for time and *j* (or *i*) points to the j^{th} (or i^{th}) sigma point; therefore χ_{k-1}^{j} is a $L\times 1$ vector representing the j^{th} sigma point at time *k-1*. The set of sigma points and their associated weights are computed as follows (see [67] for more detail):

$$\begin{aligned}
&\chi^{0} = \boldsymbol{\mu}_x, \qquad W_0^{(m)} = \frac{\lambda}{L+\lambda},\ W_0^{(C)} = \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta \\
&\chi^{j} = \boldsymbol{\mu}_x + \left(\sqrt{(L+\lambda)\Sigma_{\boldsymbol{x}}}\right)_j, \quad W_j^{(C)} = W_j^{(m)} = \frac{\lambda}{2(L+\lambda)}, j = 1,2,..., L \\
&\chi^{j+L} = \boldsymbol{\mu}_x - \left(\sqrt{(L+\lambda)\Sigma_{\boldsymbol{x}}}\right)_j,
\end{aligned} \tag{3.15}$$

where $\lambda = \alpha^2(L+\kappa) - L$ and $\left(\sqrt{(L+\kappa)\Sigma_{\boldsymbol{x}}}\right)_j$ is the j^{th} column of the matrix square root $(L+\kappa)\Sigma_x$, α is a scaling parameter which determines the spread of the sigma points, κ is a secondary scaling parameter and β is used to incorporate prior knowledge of the state distribution. To perform the prediction step, the set of *sigma points* are transformed by the process model, yielding a new set of points, $\boldsymbol{z}$, where we have:

$$\boldsymbol{z}_{k-1}^{i} = \chi_{k-1}^{i} + F[\chi_{k-1}^{i}] + B\, I_{Inj}(k) \ \ , i = 0,\ 1,\ ...,\ 2L \tag{3.16}$$

where z_{k-1}^{i} stands for the i^{th} new sigma point at time *k-1*. Now the *a priori* state estimate and the error covariance can be calculated [67] as:

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{z}_{k-1}^{i} \tag{3.17}$$

$$\Sigma_{\boldsymbol{x}_{k|k-1}} = \sum_{i=0}^{2L} W_i^{(C)} (\boldsymbol{z}_{k-1}^{i} - \hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{z}_{k-1}^{i} - \hat{\boldsymbol{x}}_{k|k-1})^{H} + \Sigma_{\boldsymbol{v}} \tag{3.18}$$

In order to perform the correction step, it is common to compute the so called *augmented sigma points*, χ^{*}, as follows:

$$\chi^*_{k-1} = \left[\hat{\boldsymbol{x}}_{k|k-1},\ \hat{\boldsymbol{x}}_{k|k-1} + \gamma\left(\sqrt{\Sigma_{\boldsymbol{x}_{k|k-1}}}\right)_i,\ \hat{\boldsymbol{x}}_{k|k-1} - \gamma\left(\sqrt{\Sigma_{\boldsymbol{x}_{k|k-1}}}\right)_i\right],\ i = 1,..., L \tag{3.19}$$

Similar to the prediction step, the set of *augmented sigma points* are transformed by the observation function to create a new set of points, z^* which can be stated as:

$$z^{*j}_{k-1} = C\ \chi^{*j}_{k-1} \tag{3.20}$$

where C is a $1 \times L$ vector. Therefore, the predicted measurement is calculated by:

$$\hat{y}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} z^{*i}_{k-1} \tag{3.21}$$

Then, the *a posteriori* state is estimated as:

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + G_{\boldsymbol{x}_k}(y_k - \hat{y}_{k|k-1}) \tag{3.22}$$

where $G_{\boldsymbol{x}_k} = \Sigma_{x_k y_k} \Sigma_{y_k}^{-1}$ is the Kalman gain with

$$\Sigma_{y_k} = \sum_{i=0}^{2L} W_i^{(C)} (z^{*i}_{k-1} - \hat{y}_{k|k-1})(z^{*i}_{k-1} - \hat{y}_{k|k-1})^T + \sigma_\varepsilon^2 \tag{3.23}$$

$$\Sigma_{x_k y_k} = \sum_{i=0}^{2L} W_i^{(C)} (\chi^{*i}_{k-1} - \hat{\boldsymbol{x}}_{k|k-1})(z^{*i}_{k-1} - \hat{y}_{k|k-1})^H \tag{3.24}$$

Finally the *a posteriori* error covariance matrix is calculated by

$$\Sigma_{\hat{\boldsymbol{x}}_{k|k}} = \Sigma_{\hat{\boldsymbol{x}}_{k|k-1}} - G_{\boldsymbol{x}_k} \Sigma_{y_k \boldsymbol{x}_k} G^H_{\boldsymbol{x}_k}.$$

3.4 State-Space Estimation Strategies

Our objective is to estimate both the hidden states, $\boldsymbol{x}=[V,m,n,h]^H$, and the system parameters, $\boldsymbol{w}=[g_{Na}, g_K, g_L]^H$ of the HH neuronal model (3.1) by using both the extended and unscented versions of the KF. To meet this objective, both the state prediction and

parameter estimation tasks need to be accomplished simultaneously. Therefore, we rewrite (3.4) as an augmented form to represent both the states and the model parameters:

$$\begin{cases} \boldsymbol{x}(k) = \boldsymbol{x}(k-1) + F\left[\boldsymbol{x}(k-1), \boldsymbol{w}(k-1)\right] + BI_{Inj}(k) + \boldsymbol{v}(k) \\ \boldsymbol{w}(k) = \boldsymbol{w}(k-1) + \boldsymbol{r}(k) \\ y(k) = C\boldsymbol{x}(k) + \varepsilon(k) \end{cases} \tag{3.25}$$

where $\boldsymbol{r}(k)$ is the parameter uncertainty which can be modeled as a zero mean Gaussian noise with covariance matrix Σ_r [67]. The above form of the state space equations including unknown parameters as a new set of dynamics and augmented states is commonly used in the literature [67]. In order to handle (3.25) where an uncertainty exists in both the parameter estimation and the state prediction, some valuable methods such as Schmidt-Kalman filter [66] and the state-dependent approach [67] have been developed. Apart from the sequential approaches (e.g., KF), some iterative methods based on maximum-likelihood (ML) approaches [74] and expectation-maximization (EM) algorithms [41] have been derived for linear models [19]. Since these algorithms are not sequential, they are only suitable for off-line applications [19]. Two other strategies of KF, joint and dual estimation, are to be developed in this chapter to simultaneously estimate the unknown states and the parameters of the HH neuronal model from the sole noisy membrane potential.

3.4.1 Joint Estimation Strategy

In the joint estimation strategy [66] and [68], the states and the parameters are combined together to form a joint state vector to be estimated through a single KF recursion. Thus, (3.25) is rewritten as

$$\begin{cases} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{w}(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}(k-1) + F\left[\boldsymbol{x}(k-1), \boldsymbol{w}(k-1)\right] + BI_{Inj}(k) + \boldsymbol{v}(k) \\ \boldsymbol{w}(k-1) + \boldsymbol{r}(k) \end{bmatrix} \\ y(k) = C\boldsymbol{x}(k) + \varepsilon(k) \end{cases} \tag{3.26}$$

or equivalently,

$$\begin{cases} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{w}(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}(k-1) + F\left[\boldsymbol{x}(k-1), \boldsymbol{w}(k-1)\right] \\ \boldsymbol{w}(k-1) \end{bmatrix} + \begin{bmatrix} B \\ \boldsymbol{0}_{3\times 1} \end{bmatrix} I_{Inj}(k) + \begin{bmatrix} \boldsymbol{v}(k) \\ \boldsymbol{r}(k) \end{bmatrix} \\ y(k) = \begin{bmatrix} C\ \boldsymbol{0}_{1\times 3} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{w}(k) \end{bmatrix} + \varepsilon(k) \end{cases} \tag{3.27}$$

3.4.2 Dual Estimation Strategy

In dual estimation strategy [66] we run two parallel filters, one on the state and the other on the parameters. In this case, it is important to note that we have two different sets of equations for the state and the parameters as expressed below.

State equation:

$$\begin{cases} \boldsymbol{x}(k) = \boldsymbol{x}(k-1) + F\left[\boldsymbol{x}(k-1), \boldsymbol{w}(k-1)\right] + BI_{Inj}(k) + \boldsymbol{v}(k) \\ y_1(k) = C\boldsymbol{x}(k) + \varepsilon(k) \end{cases} \tag{3.28}$$

Parameter equation:

$$\begin{cases} \boldsymbol{w}(k) = \boldsymbol{w}(k-1) + \boldsymbol{r}(k) \\ y_2(k) = C\left(\boldsymbol{x}(k-1) + F\left[\boldsymbol{x}(k-1), \boldsymbol{w}(k)\right] + BI_{Inj}(k)\right) + \varepsilon(k) \end{cases} \tag{3.29}$$

In the dual estimation strategy, the parameters, $\boldsymbol{w}$, are treated as a known vector within the state filter (3.28) at any given time, k, while the states, $\boldsymbol{x}$, are treated as a known vector in the parallel parameter filter (3.29). Actually, one can easily derive (3.29) by

considering that $y_2(k)$ depends on the previous value of the state vector $\boldsymbol{x}(k\text{-}1)$ and the current value of the parameter vector $\boldsymbol{w}(k)$.

3.5 Proposed Algorithms

In this Section, using the general framework of the EKF [67]and UKF [72], we will derive four algorithms based on the joint and dual estimation strategies [66] and [68]for the HH model in the following subsections.

3.5.1 Joint Extended Kalman Filtering (JEKF)

In the joint estimation strategy [67], as mentioned in Subsection 3.3.1, both the states and the parameters are estimated jointly, for which the derivative of the transition function has to be computed for both the states and the parameters which are represented by $\dot{F}_{\boldsymbol{x}}[\boldsymbol{x}(k),\boldsymbol{w}(k)]$ and $\dot{F}_{\boldsymbol{w}}[\boldsymbol{x}(k),\boldsymbol{w}(k)]$, respectively (see Appendix A for full derivation of these matrices). The linearized version of (3.27) can be written as,

$$\begin{cases} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{w}(k) \end{bmatrix} = \begin{bmatrix} I_{4\times4} + \dot{F}_{\boldsymbol{x}}[\boldsymbol{x}(k-1),\boldsymbol{w}(k-1)] & \dot{F}_{\boldsymbol{w}}[\boldsymbol{x}(k-1),\boldsymbol{w}(k-1)] \\ \boldsymbol{0}_{3\times4} & \boldsymbol{I}_{3\times3} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k-1) \\ \boldsymbol{w}(k-1) \end{bmatrix} \\ \qquad\qquad + \begin{bmatrix} B \\ \boldsymbol{0}_{3\times1} \end{bmatrix} I_{Inj}(k) + \begin{bmatrix} \boldsymbol{v}(k) \\ \boldsymbol{r}(k) \end{bmatrix} \\ y(k) = \begin{bmatrix} C & \boldsymbol{0}_{1\times3} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{w}(k) \end{bmatrix} + \varepsilon(k) \end{cases} \tag{3.30}$$

Now, the JEKF algorithm can be itemized for the HH model in three steps as follows.

JEKF Algorithm

1- Initialization:

$$\hat{\boldsymbol{x}}_{0|0} = \left[v(0), n(0), m(0), h(0), g_{Na}(0), g_K(0), g_L(0)\right]^H, \Sigma_{\boldsymbol{x}_{0|0}} = 0.01I$$

For $k \in \{1, 2, ...\}$, perform time and measurement updates.

2- Time update:

$$\hat{\boldsymbol{x}}_{k|k-1} = \begin{bmatrix} \hat{\boldsymbol{x}}_{k-1|k-1}(1:4) + F[\hat{\boldsymbol{x}}_{k-1|k-1}] \\ \hat{\boldsymbol{x}}_{k-1|k-1}(5:7) \end{bmatrix} + \begin{bmatrix} B \\ \boldsymbol{0}_{3\times1} \end{bmatrix} I_{Inj}(k)$$

$$\Sigma_{\boldsymbol{x}_{k|k-1}} = \begin{bmatrix} I_{4\times4} + \dot{F}_{\boldsymbol{x}}[\hat{\boldsymbol{x}}_{k-1|k-1}] & \dot{F}_{\boldsymbol{w}}[\hat{\boldsymbol{x}}_{k-1|k-1}] \\ \boldsymbol{0}_{3\times4} & \boldsymbol{I}_{3\times3} \end{bmatrix} \Sigma_{\boldsymbol{x}_{k-1|k-1}} \times \begin{bmatrix} I_{4\times4} + \dot{F}_{\boldsymbol{x}}[\hat{\boldsymbol{x}}_{k-1|k-1}] & \dot{F}_{\boldsymbol{w}}[\hat{\boldsymbol{x}}_{k-1|k-1}] \\ \boldsymbol{0}_{3\times4} & \boldsymbol{I}_{3\times3} \end{bmatrix}^H + \begin{bmatrix} \Sigma_v & \boldsymbol{0}_{4\times3} \\ \boldsymbol{0}_{3\times4} & \Sigma_r \end{bmatrix}$$

3- Measurement update:

$$G_{\boldsymbol{x}_k} = \Sigma_{\boldsymbol{x}_{k|k-1}} C^T \left(C\Sigma_{\boldsymbol{x}_{k|k-1}} C^T + \sigma_\varepsilon^2\right)^{-1}$$
$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + G_{\boldsymbol{x}_k}\left(y_k - C\hat{\boldsymbol{x}}_{k|k-1}\right)$$
$$\Sigma_{\boldsymbol{x}_{k|k}} = \left(I_{7\times7} - G_{\boldsymbol{x}_k} C\right)\Sigma_{\boldsymbol{x}_{k|k-1}}$$

In the JEKF algorithm, $\boldsymbol{x}$ indicates the augmented vector of the states and parameters. In particular, the first four components of vector $\boldsymbol{x}$ represent the states and the last three components represent the parameters. It is worthy to mention that the linearized form of the transition function is used just for computing the *a priori* covariance matrix rather than for the prediction step.

3.5.2 Dual Extended Kalman Filtering (DEKF)

Similar to the JEKF, we need to derive the state and parameter equations by linearizing (3.28) and (3.29), namely,

State equation:

$$\begin{cases} \boldsymbol{x}(k) = \left[\boldsymbol{I}_{4\times 4} + \dot{F}_{\boldsymbol{x}}[\boldsymbol{x}(k-1), \boldsymbol{w}(k-1)]\right]\boldsymbol{x}(k-1) + BI_{Inj}(k) + \boldsymbol{v}(k) \\ y(k) = C\boldsymbol{x}(k) + \varepsilon(k) \end{cases} \tag{3.31}$$

Parameter equation:

$$\begin{cases} \boldsymbol{w}(k) = \boldsymbol{w}(k-1) + \boldsymbol{r}(k) \\ y(k) = C\dot{F}_{\boldsymbol{w}}[\boldsymbol{x}(k-1), \boldsymbol{w}(k)]\,\boldsymbol{w}(k) + \varepsilon(k) \end{cases} \tag{3.32}$$

The DEKF algorithm can then be itemized for the HH model in five steps as follows.

DEKF Algorithm

1- Initialization:

$\hat{\boldsymbol{w}}_{0|0} = [g_{Na}(0), g_K(0), g_L(0)]^H$

$\hat{\boldsymbol{x}}_{0|0} = [v(0), n(0), m(0), h(0)]^H$,

$\Sigma_{x_{0|0}} = 0.01I, P_{w,0|0} = 0.01I$

For $k \in \{1, 2, ...\}$, perform time and measurement updates.

2- State time update:

$$\hat{\boldsymbol{x}}_{k|k-1} = \hat{\boldsymbol{x}}_{k-1|k-1} + F[\hat{\boldsymbol{x}}_{k-1|k-1}, \hat{\boldsymbol{w}}_{k-1|k-1}] + B\,I_{Inj}(k)$$

$$\Sigma_{\boldsymbol{x}_{k|k-1}} = \left(\boldsymbol{I}_{4\times 4} + \dot{F}_{\boldsymbol{x}}[\hat{\boldsymbol{x}}_{k-1|k-1}, \hat{\boldsymbol{w}}_{k-1|k-1}]\right)\Sigma_{\boldsymbol{x}_{k-1|k-1}}\left(\boldsymbol{I}_{4\times 4} + \dot{F}_{\boldsymbol{x}}[\hat{\boldsymbol{x}}_{k-1|k-1}, \hat{\boldsymbol{w}}_{k-1|k-1}]\right)^H + \Sigma_v$$

3- State measurement update:

$$G_{\boldsymbol{x}_k} = \Sigma_{\boldsymbol{x}_{k|k-1}} C^H \left(C\Sigma_{\boldsymbol{x}_{k|k-1}} C^H + \sigma_\varepsilon^2\right)^{-1}$$
$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + G_{\boldsymbol{x}_k}\left(y_k - C\hat{\boldsymbol{x}}_{k|k-1}\right)$$
$$\Sigma_{\boldsymbol{x}_{k|k}} = \left(I_{4\times 4} - G_{\boldsymbol{x}_k} C\right)\Sigma_{\boldsymbol{x}_{k|k-1}}$$

4- Parameter time update:

$$\hat{\boldsymbol{w}}_{k|k-1} = \hat{\boldsymbol{w}}_{k-1|k-1}$$

$$\Sigma_{\boldsymbol{w}_{k|k-1}} = \Sigma_{\boldsymbol{w}_{k-1|k-1}} + \Sigma_{r_{k-1}}$$

5- Parameter measurement update:

$$G_{\boldsymbol{w}_k} = \Sigma_{\boldsymbol{w}_{k|k-1}} \dot{F}_{\boldsymbol{w}}^H[\boldsymbol{x}_{k-1|k-1}, \boldsymbol{w}_{k|k-1}]$$
$$\times \left(\dot{F}_{\boldsymbol{w}}[\boldsymbol{x}_{k-1|k-1}, \boldsymbol{w}_{k|k-1}]\Sigma_{\boldsymbol{w}_{k|k-1}} \dot{F}_{\boldsymbol{w}}^H[\boldsymbol{x}_{k-1|k-1}, \boldsymbol{w}_{k|k-1}] + \sigma_\varepsilon^2\right)^{-1}$$

$$\hat{\boldsymbol{w}}_{k|k} = \hat{\boldsymbol{w}}_{k|k-1} + G_{\boldsymbol{w}_k}\left\{y_k - C\left(\hat{\boldsymbol{x}}_{k-1|k-1} + F\left[\hat{\boldsymbol{x}}_{k-1|k-1}, \hat{\boldsymbol{w}}_{k|k-1}\right] + BI_{Inj}(k)\right)\right\}$$

In the DEKF algorithm, $G_{\boldsymbol{x}_k}$ and $G_{\boldsymbol{w}_k}$ represent the Kalman gain for the vector $\boldsymbol{x}$ and that for $\boldsymbol{w}$ at time k, respectively.

3.5.3 Joint Unscented Kalman Filtering (JUKF)

As mentioned in the joint estimation strategy, the unknown parameters $\boldsymbol{w}$ are considered as the augmentation of state $\boldsymbol{x}$. Here, our objective is to apply the UKF to (3.27). The JUKF algorithm for the HH dynamical model is summarized below.

JUKF Algorithm

1- Initialization:

$$\hat{\boldsymbol{x}}_{0|0} = [v(0), n(0), m(0), h(0), g_{Na}(0), g_K(0), g_L(0)]^H,\ \Sigma_{\boldsymbol{x}_{0|0}} = 0.01I$$

For $k \in \{1, 2, ...\}$, perform time and measurement updates.

2- Calculating sigma points:

$$\chi_{k-1} = [\hat{\boldsymbol{x}}_{k-1|k-1},\ \hat{\boldsymbol{x}}_{k-1|k-1} + \gamma\left(\sqrt{\Sigma_{\boldsymbol{x}_{k-1|k-1}}}\right)_1, ...,\ \hat{\boldsymbol{x}}_{k-1|k-1} + \gamma\left(\sqrt{\Sigma_{\boldsymbol{x}_{k-1|k-1}}}\right)_L, \hat{\boldsymbol{x}}_{k-1|k-1} - \gamma\left(\sqrt{\Sigma_{\boldsymbol{x}_{k-1|k-1}}}\right)_1,$$
$$..., \hat{\boldsymbol{x}}_{k-1|k-1} - \gamma\left(\sqrt{\Sigma_{\boldsymbol{x}_{k-1|k-1}}}\right)_L]$$

Note: χ_{k-1} is a matrix whose columns are *2L+1* sigma points and $\left(\sqrt{\Sigma_{\boldsymbol{x}_{k-1|k-1}}}\right)_i$ is the i^{th} column of the square root matrix.

3- Updating time equation:

For *i=0*, …, *2L*,

$$\boldsymbol{z}^i_{k-1} = \begin{bmatrix} \chi^i_{k-1}(1:4) + F\left[\chi^i_{k-1}(1:4), \chi^i_{k-1}(5:7)\right] \\ \chi^i_{k-1}(5:7) \end{bmatrix} + \begin{bmatrix} B \\ \boldsymbol{0}_{3\times 1} \end{bmatrix} I_{Inj}(k)$$

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{z}^i_{k-1}$$

$$\Sigma_{\boldsymbol{x}_{k|k-1}} = \sum_{i=0}^{2L} W_i^{(C)} (\boldsymbol{z}^i_{k-1} - \hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{z}^i_{k-1} - \hat{\boldsymbol{x}}_{k|k-1})^H + \begin{bmatrix} \Sigma_v & \boldsymbol{0}_{4\times 3} \\ \boldsymbol{0}_{3\times 4} & \Sigma_r \end{bmatrix}$$

4- Augmented sigma points:

$$\chi^*_{k-1} = [\hat{\boldsymbol{x}}_{k|k-1},\ \hat{\boldsymbol{x}}_{k|k-1} + \gamma\left(\sqrt{\Sigma_{\boldsymbol{x}_{k|k-1}}}\right)_1, ...,\ \hat{\boldsymbol{x}}_{k|k-1} + \gamma\left(\sqrt{\Sigma_{\boldsymbol{x}_{k|k-1}}}\right)_L,\ \hat{\boldsymbol{x}}_{k|k-1} - \gamma\left(\sqrt{\Sigma_{\boldsymbol{x}_{k|k-1}}}\right)_1,$$
$$..., \hat{\boldsymbol{x}}_{k|k-1} - \gamma\left(\sqrt{\Sigma_{\boldsymbol{x}_{k|k-1}}}\right)_L].$$

For *i=0*, …, 2*L*,

$$z_{k-1}^{*i} = \left[C \;\; \boldsymbol{0}_{1\times 3} \right] \chi_{k-1}^{*i}$$

$$\hat{y}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} z_{k-1}^{*i}$$

5- Measurement updating:

$$\Sigma_{y_k} = \sum_{i=0}^{2L} W_i^{(C)} (z_{k-1}^{*i} - \hat{y}_{k|k-1})(z_{k-1}^{*i} - \hat{y}_{k|k-1})^H + \sigma_\varepsilon^2$$

$$\Sigma_{\boldsymbol{x}_k y_k} = \sum_{i=0}^{2L} W_i^{(C)} (\chi_{k-1}^{*i} - \hat{\boldsymbol{x}}_{k|k-1})(z_{k-1}^{*i} - \hat{y}_{k|k-1})^H$$

$$G_{\boldsymbol{x}_k} = \Sigma_{\boldsymbol{x}_k y_k} \Sigma_{y_k}^{-1}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + G_{\boldsymbol{x}_k} (y_k - \hat{y}_{k|k-1})$$

$$\Sigma_{\boldsymbol{x}_{k|k}} = \Sigma_{\boldsymbol{x}_{k|k-1}} - G_{\boldsymbol{x}_k} p_{y_k} G_{\boldsymbol{x}_k}^H$$

where,

$$W_1^{(m)} = \frac{\lambda}{L+\lambda}, W_i^{(m)} = \frac{\lambda}{2(L+\lambda)} \quad , \quad i = 2,3,...,2L+1$$

$$W_1^{(C)} = \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta, W_i^{(C)} = \frac{\lambda}{2(L+\lambda)} \quad , \quad i = 2,3,...,2L+1$$

In JUKF algorithm, $W^{(C)}$ and $W^{(m)}$ are the weight for calculating the mean and the covariance of the sigma points, respectively, α determines the spread of the sigma points around its mean and is usually set to a small positive value, β is used to incorporate prior knowledge of the state distribution, L (=7) is the dimension of the state ($\hat{\boldsymbol{x}}_0$), and $\gamma = \sqrt{L+\lambda}$ with λ being the composite scaling parameter.

3.5.4 Dual Unscented Kalman Filtering (DUKF)

As shown in the dual Kalman filtering strategy, the KF algorithm has been employed twice, one for the state prediction and one for the parameter estimation. The DUKF approach is described below.

DUKF Algorithm

1- Initialization:

$$\hat{\boldsymbol{w}}_{0|0} = \left[g_{Na}(0), g_K(0), g_L(0)\right]^H$$

$$\hat{\boldsymbol{x}}_{0|0} = \left[v(0), n(0), m(0), h(0)\right]^H,$$

$$\Sigma_{\boldsymbol{x}_{0|0}} = 0.01I,\ \Sigma_{\boldsymbol{w}_{0|0}} = 0.01I$$

For $k \in \{1, 2, ...\}$, perform time and measurement updates.

2- Calculating sigma points for both states and parameters:

$$\chi_{\boldsymbol{x}_{k-1}} = [\hat{\boldsymbol{x}}_{k-1|k-1},\ \hat{\boldsymbol{x}}_{k-1|k-1} + \gamma_x\left(\sqrt{\Sigma_{\boldsymbol{x}_{k-1|k-1}}}\right)_1, \ldots,\ \hat{\boldsymbol{x}}_{k-1|k-1} + \gamma_x\left(\sqrt{\Sigma_{\boldsymbol{x}_{k-1|k-1}}}\right)_L,$$
$$\hat{\boldsymbol{x}}_{k-1|k-1} - \gamma_x\left(\sqrt{\Sigma_{\boldsymbol{x}_{k-1|k-1}}}\right)_1, \ldots,\ \hat{\boldsymbol{x}}_{k-1|k-1} - \gamma_x\left(\sqrt{\Sigma_{\boldsymbol{x}_{k-1|k-1}}}\right)_L].$$

$$\chi_{\boldsymbol{w}_{k-1}} = [\hat{\boldsymbol{w}}_{k-1|k-1},\ \hat{\boldsymbol{w}}_{k-1|k-1} + \gamma_w\left(\sqrt{\Sigma_{\boldsymbol{w}_{k-1|k-1}}}\right)_1, \ldots,\ \hat{\boldsymbol{w}}_{k-1|k-1} + \gamma_w\left(\sqrt{\Sigma_{\boldsymbol{w}_{k-1|k-1}}}\right)_L,$$
$$\hat{\boldsymbol{w}}_{k-1|k-1} - \gamma_w\left(\sqrt{\Sigma_{\boldsymbol{w}_{k-1|k-1}}}\right)_1, \ldots,\ \hat{\boldsymbol{w}}_{k-1|k-1} - \gamma_w\left(\sqrt{\Sigma_{\boldsymbol{w}_{k-1|k-1}}}\right)_L].$$

Note that L_x (=4) and L_w (=3) are the states and parameters dimensions, respectively. In addition, $\gamma_x = \sqrt{L_x + \lambda_x}$ and $\gamma_w = \sqrt{L_w + \lambda_w}$ where λ_x and λ_w are the composite scaling parameters.

3- Updating time equation for state $\boldsymbol{x}$:

For $i = 0, \ldots, 2L_x$,

$$\boldsymbol{z}^{i}_{\boldsymbol{x}_{k-1}} = \chi^{i}_{\boldsymbol{x}_{k-1}} + F[\chi^{i}_{\boldsymbol{x}_{k-1}}, \hat{\boldsymbol{w}}_{k-1|k-1}] + B\, I_{Inj}(k).$$

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=0}^{2L_x} \boldsymbol{W}^{(m)}_{x,i}\, \boldsymbol{z}^{i}_{\boldsymbol{x}_{k-1}}$$

$$\Sigma_{\boldsymbol{x}_{k|k-1}} = \sum_{i=0}^{2L_x} W^{(C)}_{x,i} (\boldsymbol{z}^{i}_{\boldsymbol{x}_{k-1}} - \hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{z}^{i}_{\boldsymbol{x}_{k-1}} - \hat{\boldsymbol{x}}_{k|k-1})^{H} + \Sigma_{v}$$

4- Augmented sigma points for state $\boldsymbol{x}$:

$$\chi^{*}_{\boldsymbol{x}_{k-1}} = [\hat{\boldsymbol{x}}_{k|k-1},\ \hat{\boldsymbol{x}}_{k|k-1} + \gamma_x \left(\sqrt{\Sigma_{\boldsymbol{x}_{k|k-1}}}\right)_1, \ldots,\ \hat{\boldsymbol{x}}_{k|k-1} + \gamma_x \left(\sqrt{\Sigma_{\boldsymbol{x}_{k|k-1}}}\right)_L,$$
$$\hat{\boldsymbol{x}}_{k|k-1} - \gamma_x \left(\sqrt{\Sigma_{\boldsymbol{x}_{k|k-1}}}\right)_1, \ldots,\ \hat{\boldsymbol{x}}_{k|k-1} - \gamma_x \left(\sqrt{\Sigma_{\boldsymbol{x}_{k|k-1}}}\right)_L].$$

For $i = 0, \ldots, 2L_x$,

$$z^{*i}_{\boldsymbol{x}_{k-1}} = \left[C\ \ \boldsymbol{0}_{1\times 3}\right]\, \chi^{*i}_{\boldsymbol{x}_{k-1}}$$

$$\hat{y}_{k|k-1} = \sum_{i=0}^{2L_x} W^{(m)}_{x,i}\, z^{*i}_{\boldsymbol{x}_{k-1}}$$

5- Measurement update for state $\boldsymbol{x}$:

$$\Sigma_{y_k} = \sum_{i=0}^{2L_x} W^{(C)}_{x,i} (z^{*i}_{\boldsymbol{x}_{k-1}} - \hat{y}_{k|k-1})(z^{*i}_{\boldsymbol{x}_{k-1}} - \hat{y}_{k|k-1})^{H} + \sigma^2_{\varepsilon}$$

$$\Sigma_{\boldsymbol{x}_k y_k} = \sum_{i=0}^{2L_x} W^{(C)}_{x,i} (\chi^{*i}_{\boldsymbol{x}_{k-1}} - \hat{\boldsymbol{x}}_{k|k-1})(z^{*i}_{\boldsymbol{x}_{k-1}} - \hat{y}_{k|k-1})^{H}$$

$$G_{\boldsymbol{x}_k} = \Sigma_{\boldsymbol{x}_k y_k} \Sigma_{y_k}{}^{-1}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + G_{\boldsymbol{x}_k} (y_k - \hat{y}_{k|k-1})$$

$$\Sigma_{\boldsymbol{x}_{k|k}} = \Sigma_{\boldsymbol{x}_{k|k-1}} - G_{\boldsymbol{x}_k} \Sigma_{y_k} G_{\boldsymbol{x}_k}{}^{H}$$

6- Time update and sigma points for parameters:

$$\hat{\boldsymbol{w}}_{k|k-1} = \hat{\boldsymbol{w}}_{k-1|k-1}$$

$$\Sigma_{\boldsymbol{w}_{k|k-1}} = \Sigma_{\boldsymbol{w}_{k-1|k-1}} + \Sigma_{r_{k-1}}$$

$$\chi^{*}_{\boldsymbol{w}_{k-1}} = [\hat{\boldsymbol{w}}_{k|k-1},\ \hat{\boldsymbol{w}}_{k|k-1} + \gamma_w \left(\sqrt{\Sigma_{\boldsymbol{w}_{k|k-1}}}\right)_1, \ldots,\ \hat{\boldsymbol{w}}_{k|k-1} + \gamma_w \left(\sqrt{\Sigma_{\boldsymbol{w}_{k|k-1}}}\right)_L,$$
$$\hat{\boldsymbol{w}}_{k|k-1} - \gamma_w \left(\sqrt{\Sigma_{\boldsymbol{w}_{k|k-1}}}\right)_1, \ldots,\ \hat{\boldsymbol{w}}_{k|k-1} - \gamma_w \left(\sqrt{\Sigma_{\boldsymbol{w}_{k|k-1}}}\right)_L].$$

For $i = 0, \ldots, 2L_w$,

$$z^{i}_{\boldsymbol{w}_{k-1}} = \begin{bmatrix} C & \boldsymbol{0}_{1\times 3} \end{bmatrix} \begin{bmatrix} \left(\boldsymbol{x}_{k-1|k-1} + F\left[\hat{\boldsymbol{x}}_{k-1|k-1}, \chi^{*i}_{\boldsymbol{w}_{k-1}}\right] + BI_{Inj}(k)\right) \\ \chi^{*i}_{\boldsymbol{w}_{k-1}} \end{bmatrix}.$$

$$\hat{y}_{k|k-1} = \sum_{i=0}^{2L_w} W^{(m)}_{w,i} z^{i}_{\boldsymbol{w}_{k-1}}$$

7- Measurement update for parameters:

$$\Sigma_{y_k} = \sum_{i=0}^{2L_w} W^{(C)}_{w,i} (z^{i}_{\boldsymbol{w}_{k-1}} - \hat{y}_{k|k-1})(z^{i}_{\boldsymbol{w}_{k-1}} - \hat{y}_{k|k-1})^T + \sigma^2_{\varepsilon}$$

$$\Sigma_{\boldsymbol{w}_k y_k} = \sum_{i=0}^{2L_w} W^{(C)}_{w,i} (\chi^{*i}_{\boldsymbol{w}_{k-1}} - \hat{\boldsymbol{w}}_{k|k-1})(z^{i}_{\boldsymbol{w}_{k-1}} - \hat{y}_{k|k-1})^H$$

$$G_{\boldsymbol{w}_k} = \Sigma_{\boldsymbol{w}_k y_k} \Sigma_{y_k}^{-1}$$

$$\hat{\boldsymbol{w}}_{k|k} = \hat{\boldsymbol{w}}_{k|k-1} + G_{\boldsymbol{w}_k}(y_k - \hat{y}_{k|k-1})$$

$$\Sigma_{\boldsymbol{w}_{k|k}} = \Sigma_{\boldsymbol{w}_{k|k-1}} - G_{\boldsymbol{w}_k} \Sigma_{y_k} G_{\boldsymbol{w}_k}^{H}$$

In DUKF algorithm, $W_x^{(m)}$, $W_x^{(C)}$, $W_w^{(m)}$ and $W_w^{(C)}$ are the weights for calculating the mean and covariance of the sigma points of the state vector $\boldsymbol{x}$ and those for parameters $\boldsymbol{w}$.

It is to be noted that in DEKF and DUKF algorithms, Σ_{r_k} can be set to a small ‘‘fixed’’ diagonal matrix (e.g., $10^{-4}I$), which may then be ‘‘annealed’’ towards zero as training continues (see chapter 5 of [67]).

The performances of the various algorithms are studied in different conditions in the next section.

3.6 Performance of the Proposed Algorithms

In our simulation, an HH model with the same specifications as in [18] is used to generate the response of a spiking neuron (fast spiking) to an injected constant current. The HH model specifications are summarized in Table 3.1.

It is to be noted that all the simulations were carried out by MATLAB where the true states ($[V, n, m, h]^H$) of the HH neuron model are obtained by solving (3.1) using the “ode15” of MATLAB functions with 0.01 ms as the integration time step, while the membrane potential is sampled every 0.1 *ms*. Since this sampling rate is sufficiently high, no filter instability is observed for the KF-based algorithms. The stimulus, I_{Inj} = 0.35 μA/cm^2, with $20ms \leq t \leq 100ms$, is considered as the injected current. A zero mean white noise (considered as observation noise) with $\sigma=21.81mV$ (SNR=0 dB) is added to the generated membrane voltage. The dynamic noise variance for $[V, n, m, h]^H$ is respectively set to $[0.01mV, 1e\text{-}4, 1e\text{-}4, 1e\text{-}4]^H$.

In order to accomplish a fair comparison, the initial values of the model parameters, g_K, g_{Na}, g_L, are the same as those in [18] which are slightly different from their actual values.

Our objective is to verify the feasibility of each of the aforementioned KF approaches to estimate the hidden states and unknown parameters of the above HH model from the noisy membrane potential as shown in Figure. 3.2.

Table 3.1: HH model specifications

Symbol	***Specification***	***Description***
E_L	-54.387 mV	Leakage reversal potential
E_{Na}	50 mV	Sodium Reverse Potential
E_K	-77 mV	Potassium Reverse Potential
C_M	1 $\mu F/cm^2$	Membrane capacitance
g_L	0.3 mS/cm^2	Leakage Conductance
g_{Na}	120 mS/cm^2	Sodium Conductance
g_K	36 mS/cm^2	Potassium Conductance

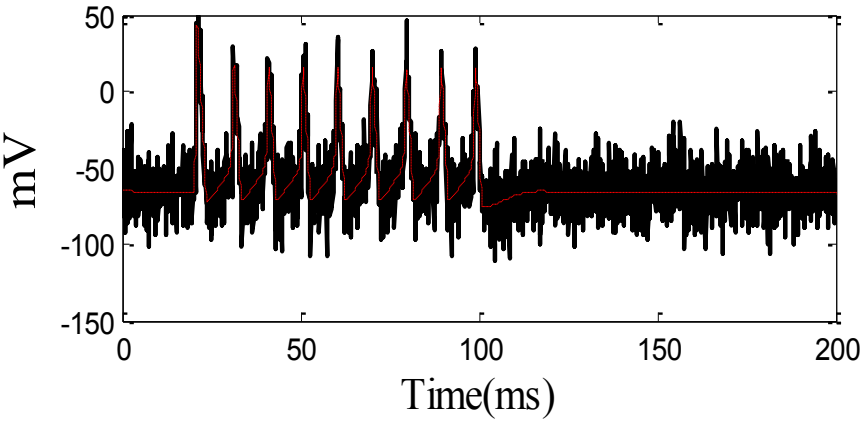
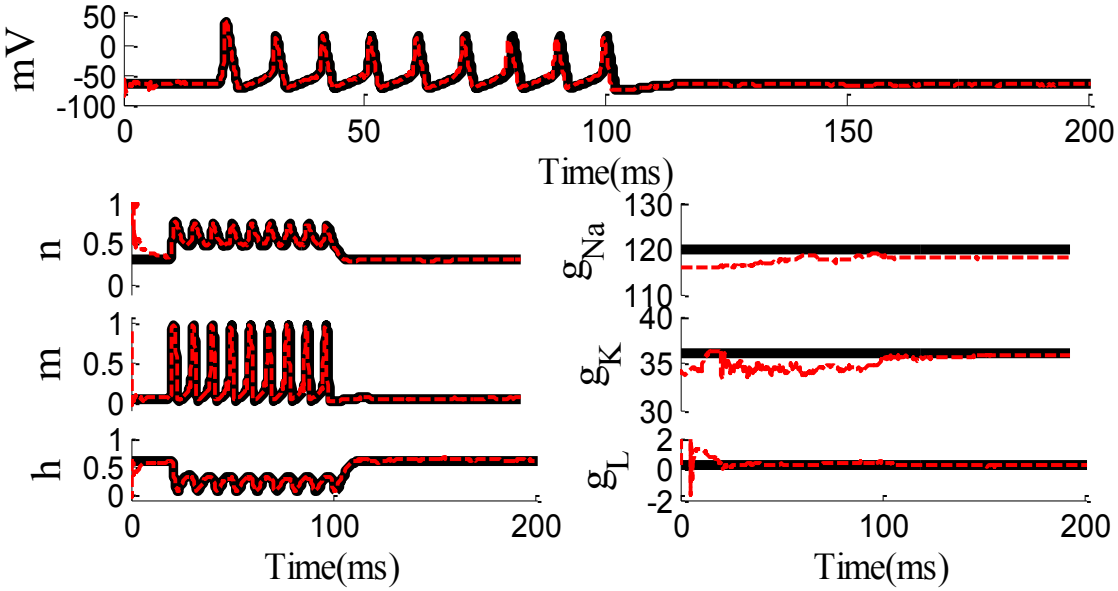


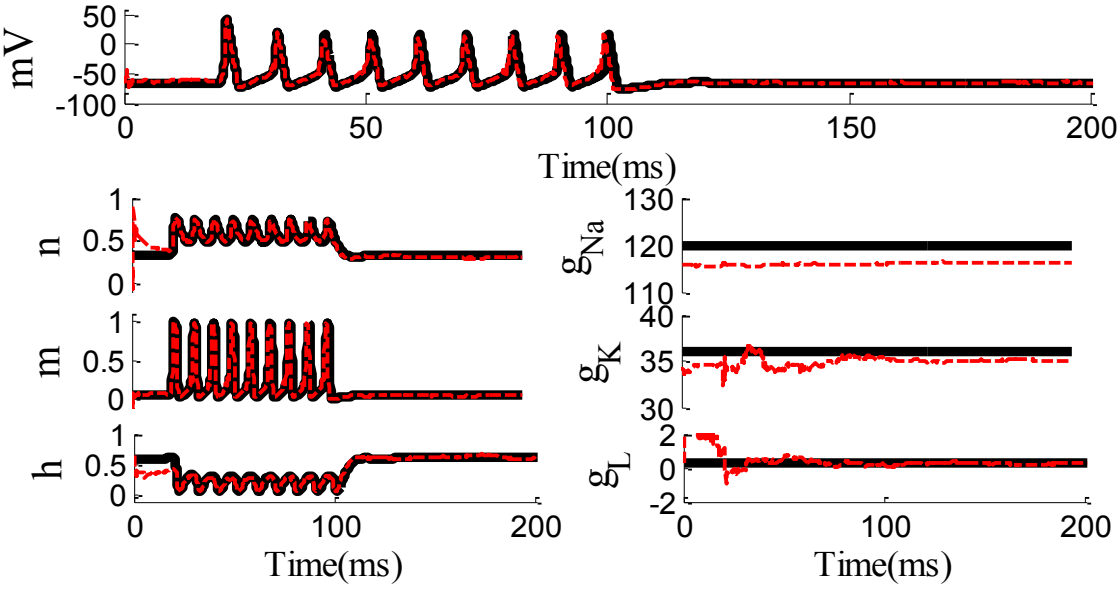
Figure 3.2. Noisy observed membrane voltage (black solid line) versus the true voltage (red dashed line).

Figure 3.3 (A-D) presents, respectively, the performances of JUKF, DUKF, JEKF and DEKF in tracking the dynamics and estimating the parameters of the simulated HH model.

JEKF


mV
50
0
-50
-100
0
50
100
150
200
Time(ms)
n
m
h
1
0.5
0
0
100
200
Time(ms)
g_{Na}
g_K
g_L
130
120
110
40
35
30
2
0
-2
0
100
200
Time(ms)

DEKF


mV
50
0
-50
-100
0
50
100
150
200
Time(ms)
n
m
h
1
0.5
0
0
100
200
Time(ms)
g_{Na}
g_K
g_L
130
120
110
40
35
30
2
0
-2
0
100
200
Time(ms)

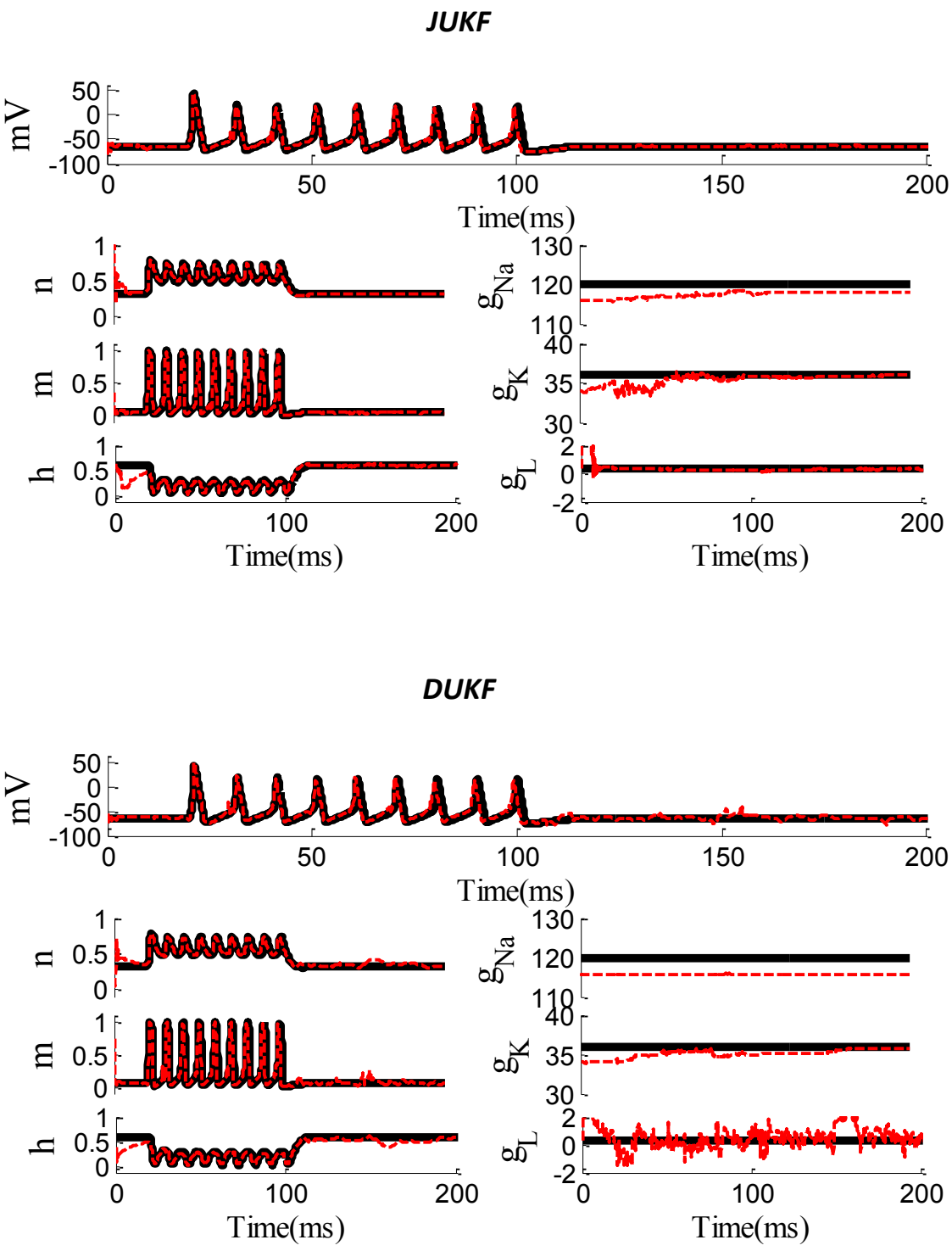


Figure 3.3. Tracking the dynamics of HH neuron model using A) JUKF, B) DUKF, C) JEKF), D) DEKF. For each method, voltage V, K^+ channel activation n, Na^+ channel activation m, Na^+ channel inactivation h, g_{Na}, g_K and g_L are drawn. Black solid and red dashed lines indicate the true (original) and the estimated values. The observation noise standard deviation $\sigma=21.81mV$ (SNR=0 dB).

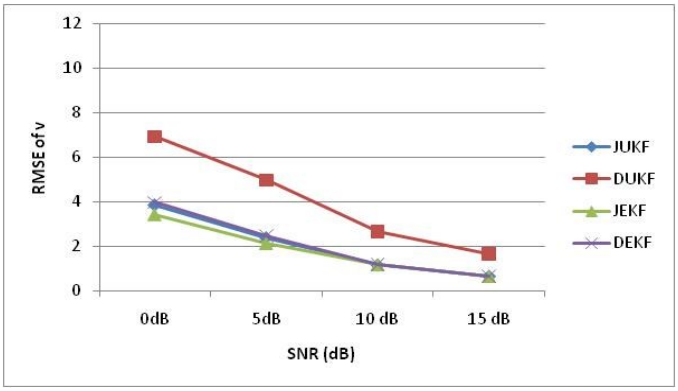
It is clear from Figure 3.3 that the intracellular voltage (V), the hidden states (m, n and h) and the unknown parameters (g_{Na}, g_K and g_L) have been reconstructed with excellent accuracy from the noisy observation. These figures demonstrate the promising performance of each method in estimating the states, [V, m, n, h], of the HH model when the initial values of parameters are sufficiently close to their true values. However, it is observed that g_K and g_L are always better estimated (converged) than g_{Na}.

In order to make a more precise comparison between these methods, we perform the second part of our simulation to investigate the performance of these algorithms at different levels of the observation noise (0-15 dB) by computing the root mean square error (RMSE) between the estimated variables and the corresponding true values. It is to be noted that initial values of the parameters, in this case, are randomly selected from the ±25% neighborhood of the true values. This prior assumption about the parameter's boundary is consistent with the results of [30] to ensure the identifiability of the HH model. The SNR ratio is calculated as below:

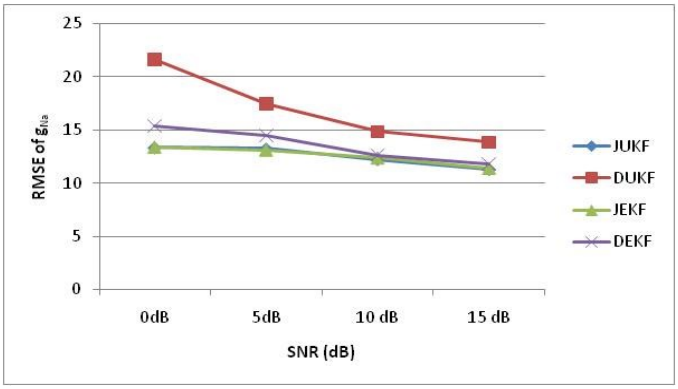
$$SNR_{dB} = 10\log_{10}\frac{E\{(V - E\{V\})^2\}}{\sigma^2}$$

where $E\{.\}$ stands for the expected value and σ^2 is the observation noise variance. It is noteworthy that though the measurement noise in patch clamp or dynamic clamp is quite small but here, similar to [21], our scope is to analyze the accuracy of the proposed algorithms even at high level of observation noise. Figure 3.4 (A-D) is drawn to show the RMSE of the membrane voltage, v, and the maximum conductances, [g_{Na}, g_K, g_L]. As seen in Figure 3.4 (A-D), JUKF, JEKF and DEKF lead to better results in smoothing noisy membrane potential as well as estimating the maximum conductances than DUKF, especially at low SNR. While the sensitivity analysis of the maximum conductances in

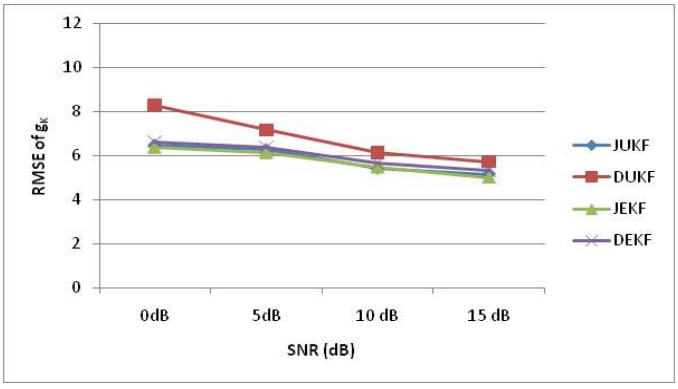
the HH neuronal model is nontrivial [30], we aim here to calculate the estimation accuracy of these parameters for different SNRs.



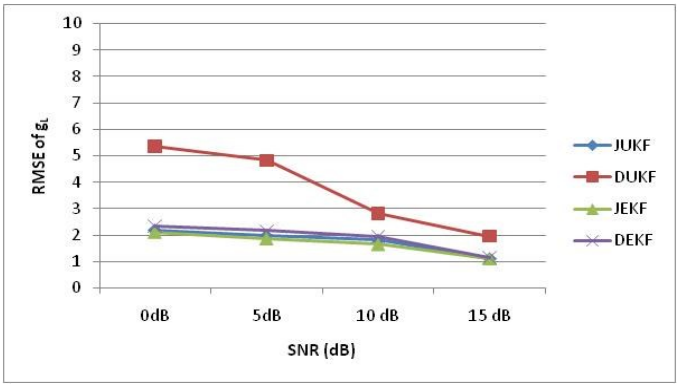
(a)



(b)



(c)



(d)

Figure 3.4. RMSE versus SNR for (a) V, (b) g_{Na}, (c) g_K and (d) g_L.

It is found that RMSE of g_K and g_L does not change considerably for different SNRs (except for JUKF algorithm); further, changes in the RMSE of g_{Na} for different SNRs are more significant than those of g_K and g_L. Considering the RMSE of all maximum conductances as well as the membrane potential of the HH model, DUKF can perform as equally well as the other methods, when either the initial value of parameters are close to their true values (as we can see in Figure 3.3 except for the g_{Na}) or the SNR is fairly high. Although the difference between JUKF, JEKF and DEKF is very small, JUKF and JEKF are more robust to the observation noise. Moreover, Figure 3.5 corresponds to the case when the initial values of the HH model parameters are not selected from the mentioned boundary (±25% of their true values). Although this example is an extreme case, it confirms that the KF-based algorithms still perform accurately. The estimated g_{Na} is biased for all the algorithms, this is due to the observation noise because of which the algorithms underestimate the sharpness of spikes, and therefore underestimates the sodium current.

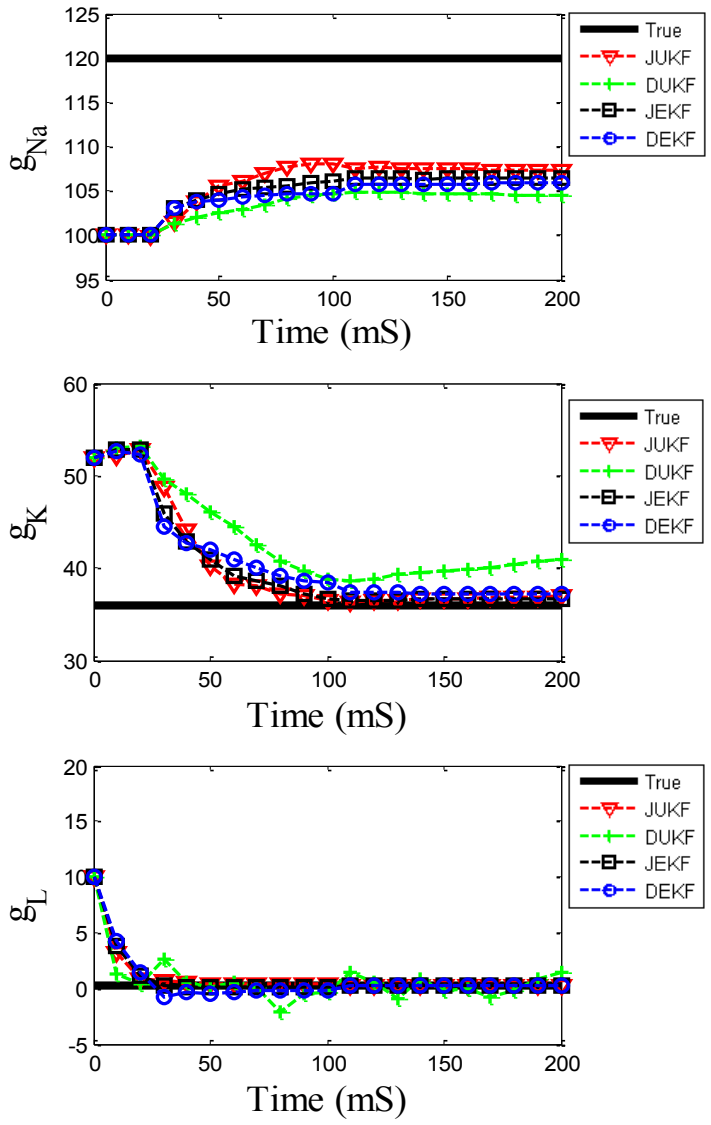


Figure 3.5. Performance of proposed methods in estimating the parameters of the HH model for SNR = 10 dB. The color map is drawn on the right hand side of each panel.

The results of our other simulations, with different specifications for the HH model, confirm that the joint estimation strategy of the KF method performs better than the dual strategy especially for the UKF. The performances of JEKF and JUKF are almost always the same in all simulations. Furthermore, it is worth mentioning that JEKF is much faster than JUKF.

In order to provide a statistical analysis of the performance of each of the algorithms in estimating the maximum conductances (g_{Na}, g_K and g_L), another simulation including 100 trials, each lasting 200 ms, is conducted. At each trial, an Ornstein-Uhlenbeck (O-U) process (colored noise is filtered by $0.4/(1-0.9z^{-1})$) is used as the stimulus, I_{stim} (HH model specification is the same as table 3.1). The results for each algorithm (over 100 trials) are summarized in Table 3.2.

Table 3.2: Statistical analysis of four mentioned algorithms. *Mean* ± *std* of each parameter over different trials is shown.

Method\parameter	g_{Na}	g_K	g_L
JEKF	117.12±5.75	36.23±2.02	0.3±0.04
JUKF	117.14±5.64	35.87±2.13	0.3±0.04
DEKF	117.33±6.11	35.82±2.13	0.3±0.04
DUKF	113.86±9.14	32.73±4.47	0.29±0.17

As can be seen from this table, the estimates of g_K and g_L are unbiased (except for the DUKF), while the estimated g_{Na} is biased for all the algorithms. Moreover, it is observed that the extended versions of the KF, both the joint and dual estimation strategies, as well as JUKF give approximately the same results in estimating both the parameters and the dynamics of the HH model.

3.7 Discussion

In addition to the above mentioned observations made from the results of the simulations, we now consider the performance of each of the KF algorithms from a signal processing point of view. EKF suffers, in general, from two drawbacks [75]. First, the derivation of the Jacobian matrices used for linearizing the nonlinear process and the observation

functions are computationally complex and difficult to implement. Second, this linearization may lead to filter-instability, if the sampling time is not sufficiently small. The first issue of EKF in the HH neuronal model is addressed in this thesis by analytically computing such matrices (see Appendix A). It is also easy to extend this procedure for the HH models with more dynamics such as calcium, and extracellular potassium concentration [18]. The second issue, namely, the sampling period, can be neglected since in electrophysiological recordings the sampling frequency is sufficiently high (about10 KHz - 25 KHz). Although UKF does not suffer from such issues, it is not always a better alternative than EKF in dealing with nonlinear systems. As an example, the authors of [75] conducted an empirical comparison between UKF and EKF when they are applied to human motion tracking for virtual reality in the presence of noise. Both the analysis and experimental results in [75] indicated that UKF performs equally well as EKF does. Similarly in our simulation, the UKF and EKF demonstrated approximately equal performance especially in low SNRs. The authors of [75] reported that EKF is a better choice for estimating quaternion motion in virtual reality applications. Another important comparison between EKF and UKF algorithms in this thesis can be accomplished by taking the difference of the joint and dual KF strategies into account. This difference arises from the statistical dependency of the states, [v, n, m, h], and the parameters, [g_{Na}, g_K, g_L], of the HH neuronal model in the joint estimation strategy, but not in the dual estimation strategy. In other words, the joint estimation technique allows explicit computation of the cross covariance of the states and the parameters of the HH model which is, however, zero in the dual estimation technique. On the other hand, the experiments performed in [76] show little difference between the two approaches. In

conclusion, EKF approaches, both the joint and the dual estimation strategies, demonstrate promising performance in tracking the dynamics and estimating the parameters of the HH model. One of the most important issues of tracking the HH model's dynamics arise from the high level of the dynamical noise (process noise) rather than the observation noise. The authors of [18] and [37] selected the noise covariance matrices very carefully to yield the best results. However, in our simulation studies, although the variance of the dynamical noise is negligible, precise selection of the noise covariance matrix plays an important role in the convergence of the parameters and the dynamics. We believe that using adaptive algorithms such as [77], which iteratively update noise covariance matrices can significantly improve the performance of both the UKF and EKF based algorithms and make them more applicable even for HH models with high dynamical noise i.e., high level of uncertainties in the HH dynamics.

The speed of the designed algorithms is of significant importance especially in some applications such as dynamic clamp [12], where the time required for reading the membrane potential and calculating the current to inject has to be less than the shortest time constant in a real neuron. This limitation can be problematic when the biophysical model is computationally intensive such as the HH model. Considering the problem we have addressed here, where there are seven unknown variables (15 sigma points), UKF is considerably faster than the particle filtering method in [21]. Moreover, the EKF-based algorithms (JEKF and DEKF) run faster than the unscented ones. One can conclude that UKF takes longer computation time than EKF because it has to handle all the sigma points. Therefore, the feature of fast implementation of the EKF approaches can compensate for the major limitation of the dynamic clamp technique. Furthermore, the

results of our simulations indicate that the performance of the EKF approaches is equivalent to that of the JUKF, although the EKF-based methods are much faster and therefore are more applicable in real time applications.

3.8 Identification of Entire Ion Channels Kinetics of HH Neuronal Model

Estimating the parameters and tracking the dynamics of the HH neuronal model have been done in Sections 3.2 – 3.5. The algorithms proposed therein have been developed for a HH model with known kinetics[5] [78, 79]. These kinetics, e.g., in dynamic clamp technique, can be measured experimentally [12], and therefore in this technique, as an example, tracking the hidden dynamics (ion channels) and estimating the maximum conductances of a single neuron from the recorded membrane potential is of the core interest. In such cases, i.e., neurons with known kinetics, our previously proposed algorithms, namely, JUKF, DUKF, JEKF and DEKF, are suitable to be employed. However, in some other applications, e.g., spike prediction [80], the objective is to find the best neuronal model that predicts the spike timing of a single neuron [81]. It is to be noted that simpler neuronal models have been widely used in the literature to address spike timing prediction. As mentioned in Sections 1.2 and 3.1, the HH neuronal model is the most biophysically detailed model that can properly mimic behavior of the spiking neurons. The critical question is why the HH neuronal model is rarely employed for spike timing prediction [81] (see also [25]). The answer relies on the complexity and highly

[5] Ion channels' kinetics are referred to the gating variables in the dynamics of ion channels.

nonlinear structure of the HH model. Generally speaking, no prior knowledge about the ion channels kinetics of the HH model is available, and estimating the entire kinetics is nontrivial [81]. In order to apply the HH model to predict spikes of a neuron, all of its kinetics and intrinsic parameters, including the maximum conductances and the reversal potentials, must be inferred from the recorded membrane potential only. Successful methods for estimating the entire parameters of the HH neuronal model use several trials of different types of the injected currents in conjunction with the optimization based techniques (see [24] and references therein). However, in the category of Bayesian approaches where only one trial of the injected current is available, the particle filtering [21] is the only work in the literature that estimates the entire set of the HH model parameters.

Our objective in this section is to extend our proposed algorithms in order to estimate the entire parameters of the HH neuronal model. Estimating such parameters provides better understanding about the dynamics of the ion channels. These hidden dynamics can be reconstructed from the noisy membrane potential. In this part, according to the appropriate performances of the JUKF and JEKF in our numerical simulations presented in Section 3.6, we develop these algorithms to estimate the entire set of the HH model parameters.

Using (2.8), (2.9) and (2.10), and considering the entire parameters of the HH model to be unknown, the state and parameter vectors are respectively defined as: $\boldsymbol{x} = [V, n, m, h]$, $\boldsymbol{w} = [g_{Na}, g_K, g_L, E_{Na}, E_K, E_L, V_{th(n)}, V_{th(m)}, V_{th(h)}, s_n, s_m, s_h, t_n, t_m, t_h]$. All these variables have been already introduced in Section 2.3. According to these definitions for the state

and parameter vectors of the HH neuronal model, and in consistent with [21], the dynamics of the ion channels are expressed by the Langevin equation [82] as follows.

$$\frac{dn}{dt} = \frac{n_\infty(V) - n}{\tau_n(V)},\ \frac{dm}{dt} = \frac{m_\infty(V) - m}{\tau_m(V)},\ \frac{dh}{dt} = \frac{h_\infty(V) - h}{\tau_h(V)} \tag{3.33}$$

where $n_\infty(V)$, $m_\infty(V)$ and $h_\infty(V)$ represent the steady state values of the gating variables n, m and h, respectively, and $\tau_n(V)$, $\tau_m(V)$ and, $\tau_h(V)$ are the corresponding time constants (see Section 2.3 for more details). Considering (3.1) and (3.33), the state space representation of the HH neuronal model including the entire set of parameters can be stated as follows.

$$\begin{cases} \dfrac{d\boldsymbol{x}(t)}{dt} = \Gamma[\boldsymbol{x}(t)] + BI_{Inj}(t) + \boldsymbol{v}(t) \\ y(t) = C\boldsymbol{x}(t) + \varepsilon(t) \end{cases} \tag{3.34}$$

where $\Gamma[\boldsymbol{x}(t)]$ is the transient function given by

$$\Gamma[\boldsymbol{x}(t)] = \begin{bmatrix} \Gamma_1[\boldsymbol{x}(t)] \\ \Gamma_2[\boldsymbol{x}(t)] \\ \Gamma_3[\boldsymbol{x}(t)] \\ \Gamma_4[\boldsymbol{x}(t)] \end{bmatrix} = \begin{bmatrix} \dfrac{-g_{Na}m^3h(V-E_{Na}) - g_K n^4(V-E_K) - g_L(V-E_L)}{C_M} \\ (n_\infty(V) - n)/\tau_n(V) \\ (m_\infty(V) - m)/\tau_m(V) \\ (h_\infty(V) - h)/\tau_h(V) \end{bmatrix} \tag{3.35}$$

where $B = [1/C_M, 0_{1\times3}]^{\mathrm{H}}$ and C = [1, $0_{1\times3}$]. Note that the transient function $F[\boldsymbol{x}(t)]$ in (3.2) was defined for a known kinetics, but in general, when the kinetics are unknown, (3.34) and (3.2) are equivalent. Furthermore, (3.30), the equation from which JEKF has been derived, can be written as:

$$\begin{cases} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{w}(k) \end{bmatrix} = \begin{bmatrix} I_{4\times4} + \dot{\Gamma}_{\boldsymbol{x}}[\boldsymbol{x}(k-1), \boldsymbol{w}(k-1)] & \dot{\Gamma}_{\boldsymbol{w}}[\boldsymbol{x}(k-1), \boldsymbol{w}(k-1)] \\ \boldsymbol{0}_{15\times4} & \boldsymbol{I}_{15\times15} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k-1) \\ \boldsymbol{w}(k-1) \end{bmatrix} \\ \qquad + \begin{bmatrix} B \\ \boldsymbol{0}_{15\times1} \end{bmatrix} I_{Inj}(k) + \begin{bmatrix} \boldsymbol{v}(k) \\ \boldsymbol{r}(k) \end{bmatrix} \\ y(k) = \begin{bmatrix} C & \boldsymbol{0}_{1\times15} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{w}(k) \end{bmatrix} + \varepsilon(k) \end{cases} \tag{3.36}$$

where the derivative of the transition function with respect to the states, $\dot{\Gamma}_{\boldsymbol{x}}[\boldsymbol{x}(k), \boldsymbol{w}(k)]$, and the parameters, $\dot{\Gamma}_{\boldsymbol{w}}[\boldsymbol{x}(k), \boldsymbol{w}(k)]$, are calculated in Appendix A.

And, (3.27), the equation from which JUKF has been derived, can be written as:

$$\begin{cases} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{w}(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}(k-1) + \Gamma[\boldsymbol{x}(k-1), \boldsymbol{w}(k-1)] \\ \boldsymbol{w}(k-1) \end{bmatrix} + \begin{bmatrix} B \\ \boldsymbol{0}_{15\times1} \end{bmatrix} I_{Inj}(k) + \begin{bmatrix} \boldsymbol{v}(k) \\ \boldsymbol{r}(k) \end{bmatrix} \\ y(k) = \begin{bmatrix} C & \boldsymbol{0}_{1\times15} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{w}(k) \end{bmatrix} + \varepsilon(k) \end{cases} \tag{3.37}$$

Now, JEKF and JUKF, in the same manner described in subsections 3.5.1 and 3.5.2, can be applied to (3.36) and (3.37), respectively, to track the hidden dynamics, $\boldsymbol{x}$, and infer the entire parameters, $\boldsymbol{w}$, of the HH neuronal model. The results of these algorithms for estimating the entire parameters of the HH model will be shown in the next section.

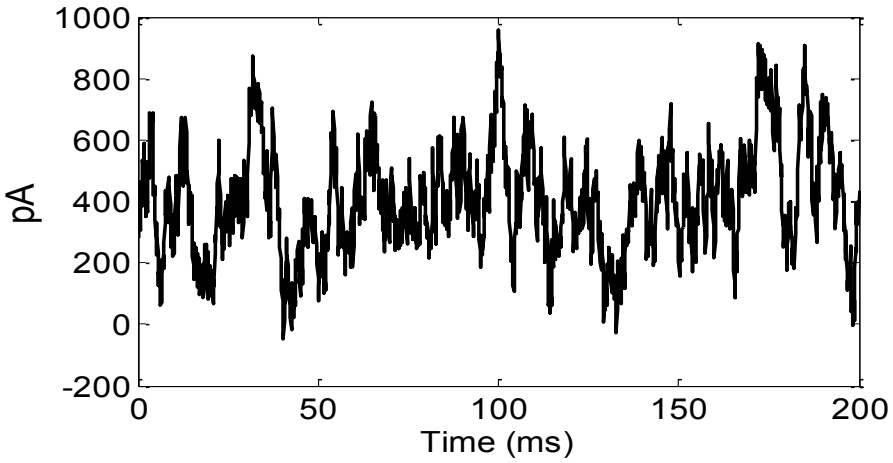
3.9 Performance of JUKF and JEKF for Estimating Entire Parameters of HH Model

In this part, the developed JEKF and JUKF algorithms for estimating the entire parameters of the HH neuronal model are verified in two experiments. In the first

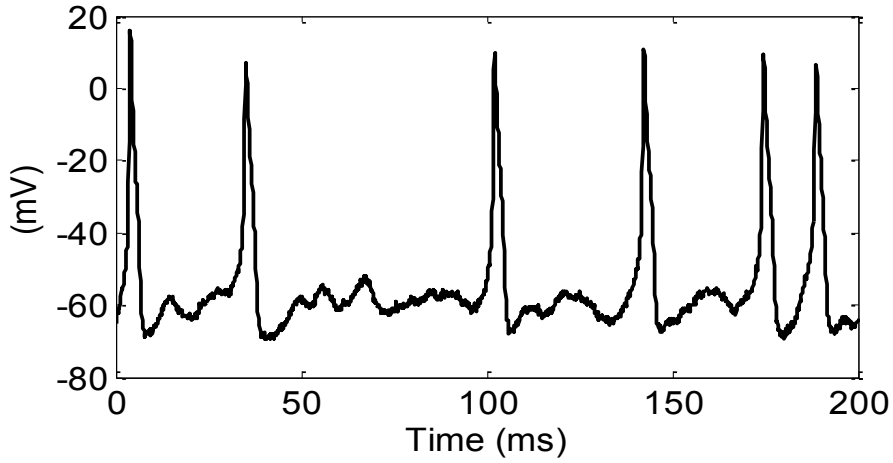
experiment, the accuracy of these algorithms is checked within a simulation study. In the second experiment, the real data recordings from a single neuron are used to test the performance of the proposed algorithms. Before conducting these experiments, several points are to be clarified. There are some protocols for recording data from a single neuron using electrophysiological techniques. The most computational methods in the literature are developed for specific protocols [83]. In this thesis, due to the limited availability of such specific data recordings, our proposed methods are applied to estimate the entire HH model parameters of a regular spiking L5 pyramidal cell responded to an *in-vivo*-like current injection. The responses of this cell are recorded in current clamp mode. This data is provided from [84] wherein one can find all the instructions about the data recording procedures. Our main objective here is to verify the feasibility of the HH neuronal model to describe the spiking behavior of this neuron. To meet this objective, as mentioned before, the entire set of the HH model parameters has to be identified. To our best knowledge, this is the first time that both the unscented and extended KF algorithms are developed for the HH model to infer all its parameters.

3.9.1 Numerical Simulation

Here, a HH model with the parameters summarized in Table 3.3 is used to generate the response of a spiking neuron (fast spiking) to an injected current. This current is selected from the last 42.5 *sec* stimulus used in [84]. 200 *ms* of this stimulus (25 – 25.2 *sec*) and the corresponding generated membrane potential are shown in Figure 3.6.



(a)



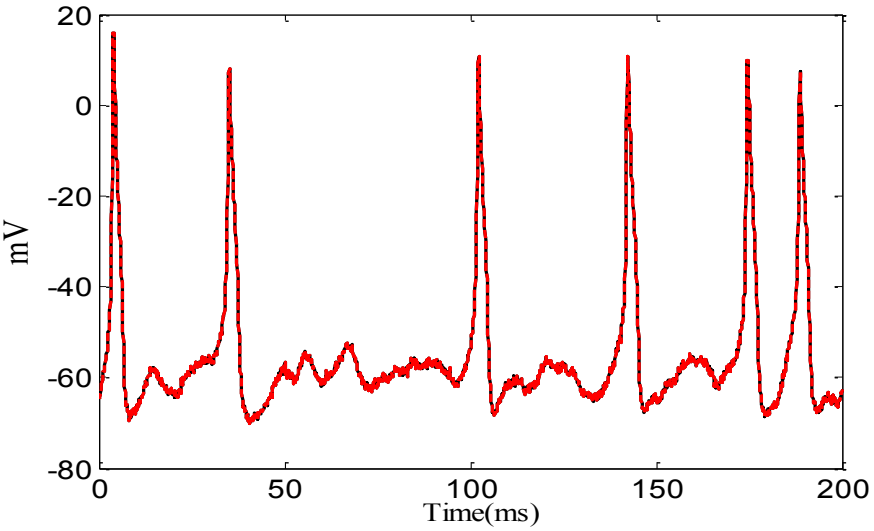
(b)

Figure 3.6. Injected current (a) and membrane potential (b) of a neuron described in Table 3.3.

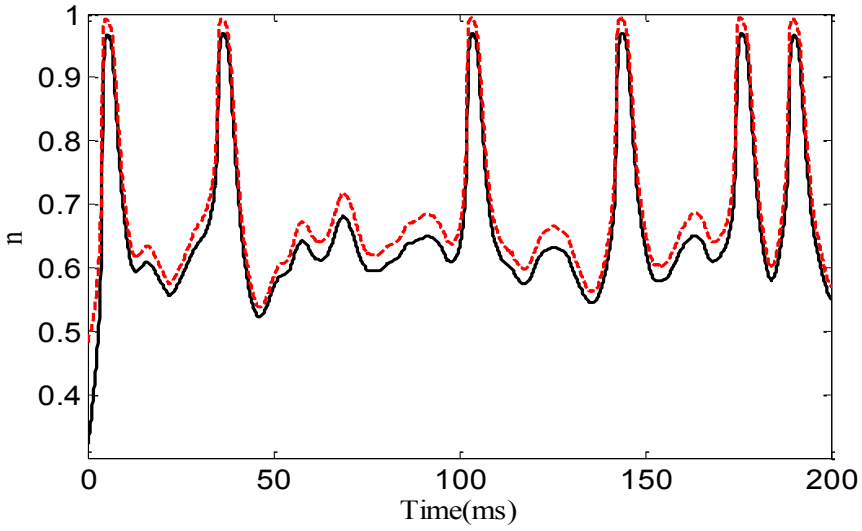
Similar to Section 3.6, all the simulations were carried out by MATLAB where the true states ($[V, n, m, h]^{H}$) of the HH neuron model are obtained by solving (3.35) using the "ode15" of MATLAB functions with 0.01 ms as the integration time step, while the membrane potential is sampled every 0.1 *ms*. A zero mean white noise (considered as observation noise) with $\sigma=1.28mV$ is added to the generated membrane voltage. The initial values of the HH model parameters are mentioned in Table 3.3. Our objective is to

verify the feasibility of the JUKF and JEKF algorithms to estimate the entire HH model parameters from the noisy membrane potential.

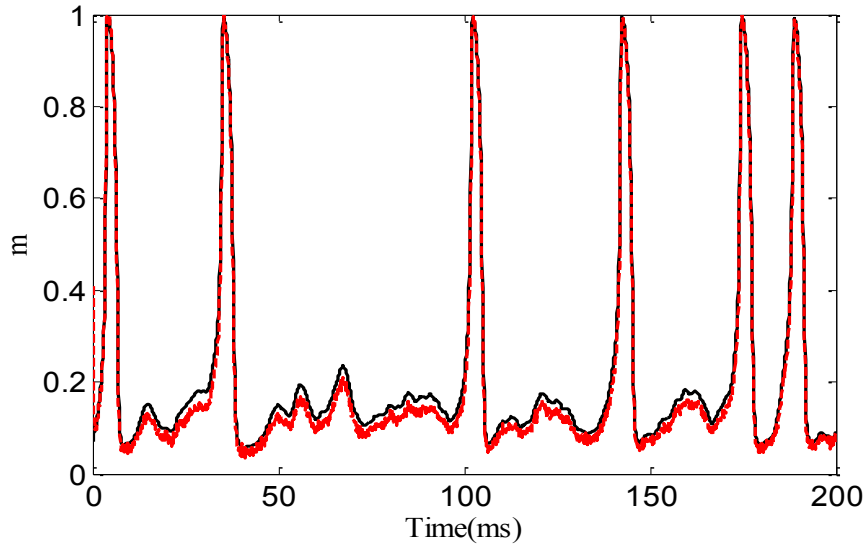
Figure 3.7 shows the results of the estimated states [V, n, m, h] of the HH model using JUKF. Moreover, other estimated parameters including the maximum conductances, the reversal potentials and the kinetic of the ion channels are presented in Table 3.3.



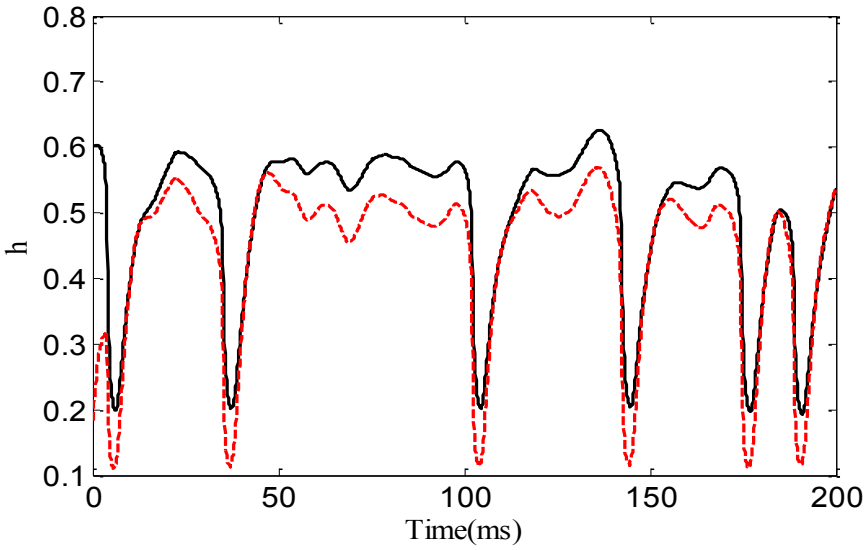
(a)



(b)



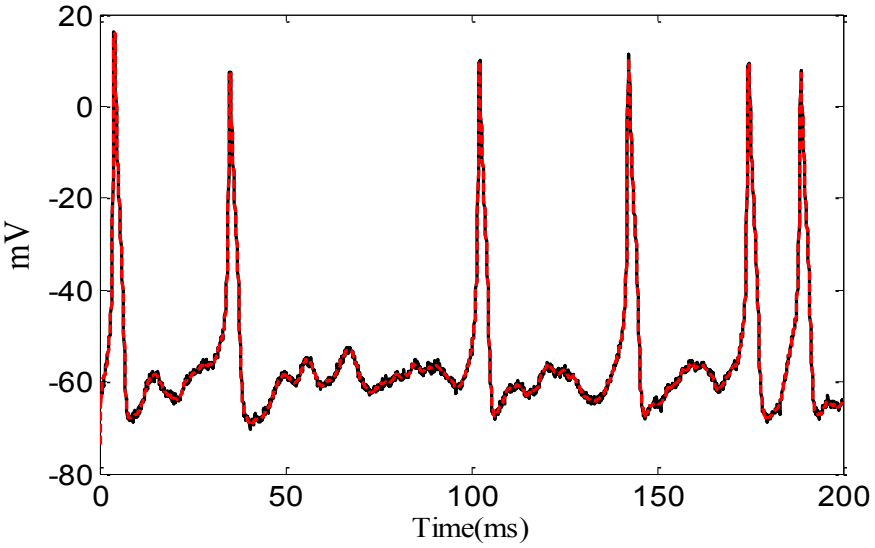
(c)



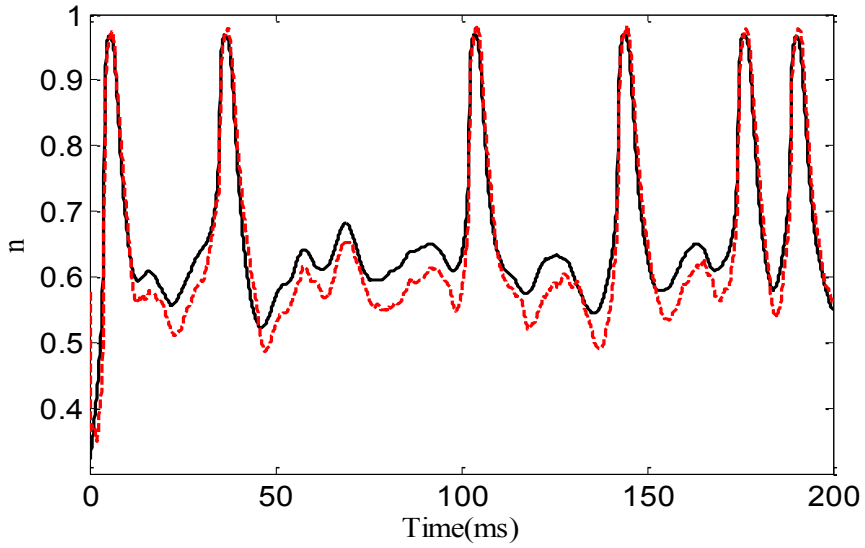
(d)

Figure 3.7. Estimated (dashed red lines) versus true (black sold line) dynamics of the HH neuronal model using JUKF, (a) membrane potential, (b) *n*, (c) *m* and (d) *h*.

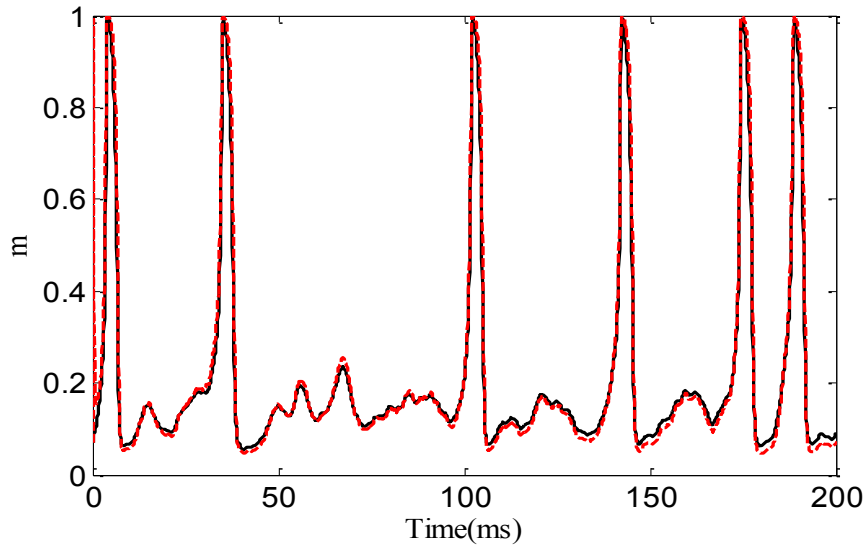
Figure 3.8 shows the estimated states of the HH model using JEKF. The inferred parameters are also summarized in Table 3.3.



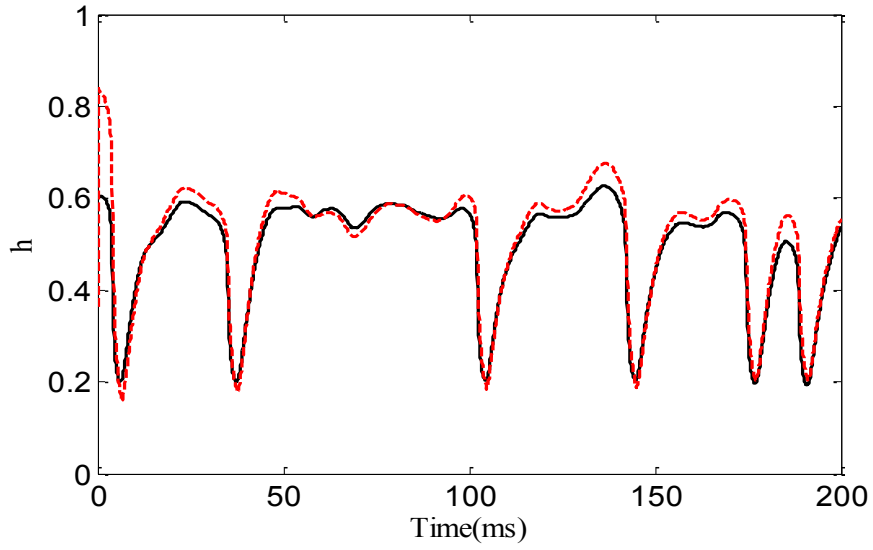
(a)



(b)



(c)



(d)

Figure 3.8. Estimated (dashed red lines) versus true (black sold line) dynamics of the HH neuronal model using JEKF, (a) membrane potential, (b) *n*, (c) *m* and (d) *h*.

Table 3.3. HH model parameters and the estimated values

Parameter	Unit	Initial value	Estimated value		True value
			JEKF	JUKF	
g_{Na}	mS/cm^2	30	13.38	15.57	12
g_K	mS/cm^2	10	3.89	2.24	3
g_L	mS/cm^2	0.5	0.12	0.02	0.1
E_{Na}	mV	60	36.68	41.57	40
E_K	mV	-80	-72.81	-73.10	-72
E_L	mV	-70	-65.07	-63.22	-60
$V_{th(n)}$	mV	-70	-63.36	-65.02	-65
$V_{th(m)}$	mV	-45	-40.69	-38.73	-41
$V_{th(h)}$	mV	-65	-52.98	-58.51	-55
S_n	-	1	0.06	0.10	0.08
S_m	-	1	0.09	0.09	0.1
S_h	-	1	-0.08	-0.07	-0.08
t_n	$msec$	10	2.49	4.44	4
t_m	$msec$	1	0.29	0.25	0.25
t_h	$msec$	10	8.54	4.89	8

The inferred hidden dynamics using both JUKF and JEKF (see Figures 3.7 and 3.8) are nicely tracking their true values. Moreover, as can be seen from Table 3.3, all parameters of the HH neuronal model are estimated in a good interval of their true values using both JUKF and JEKF algorithms. In order to check whether the estimated parameters can

mimic the spiking behavior of the neuron (spike timing prediction), these parameters are used to reconstruct the membrane potential of the neuron. The reconstructed membrane potential using JUKF and JEKF are plotted in Figures 3.9 and 3.10, respectively.

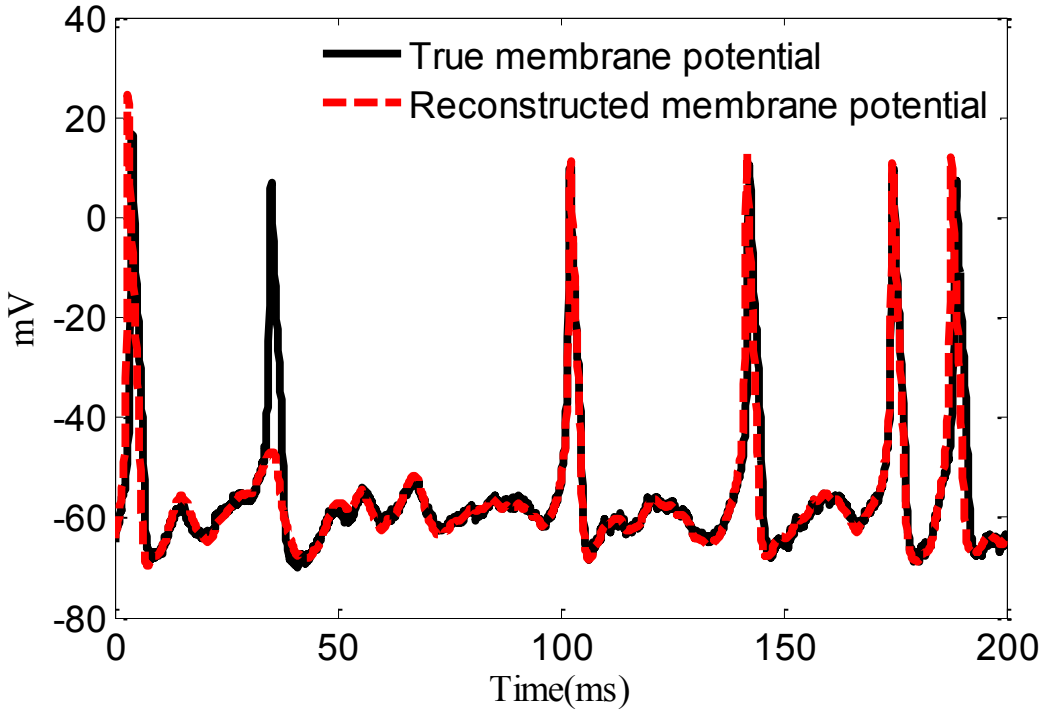


Figure 3.9. Reconstructed membrane potential (red dashed line) using JUKF versus true membrane potential (solid black line).

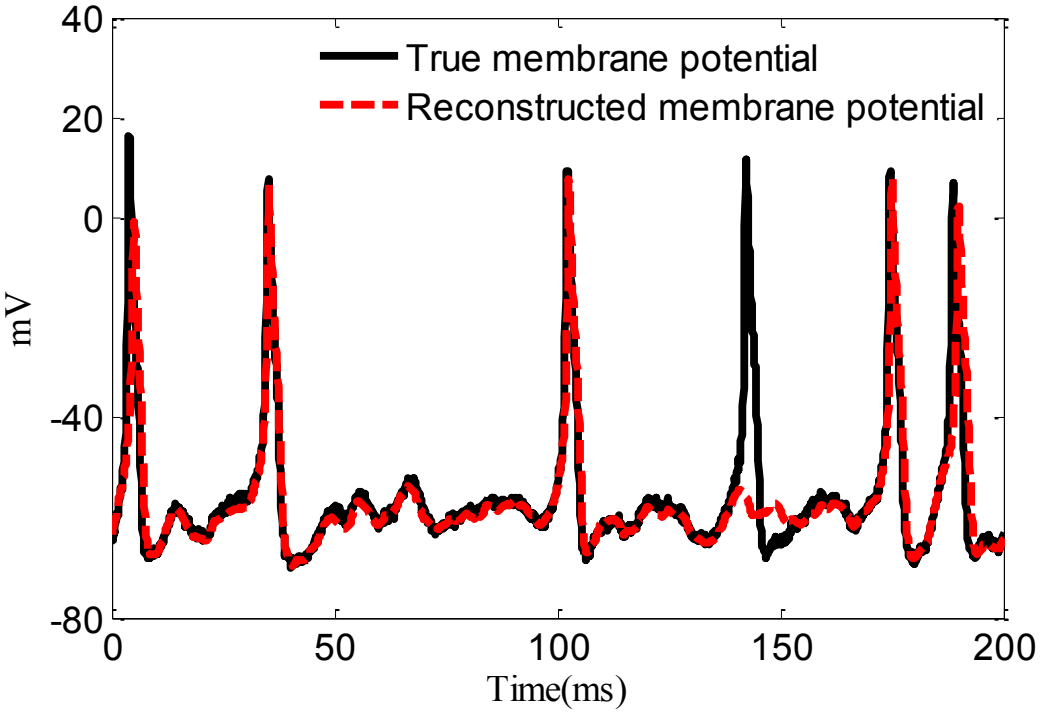


Figure 3.10. Reconstructed membrane potential (red dashed line) using JEKF versus true membrane potential (solid black line).

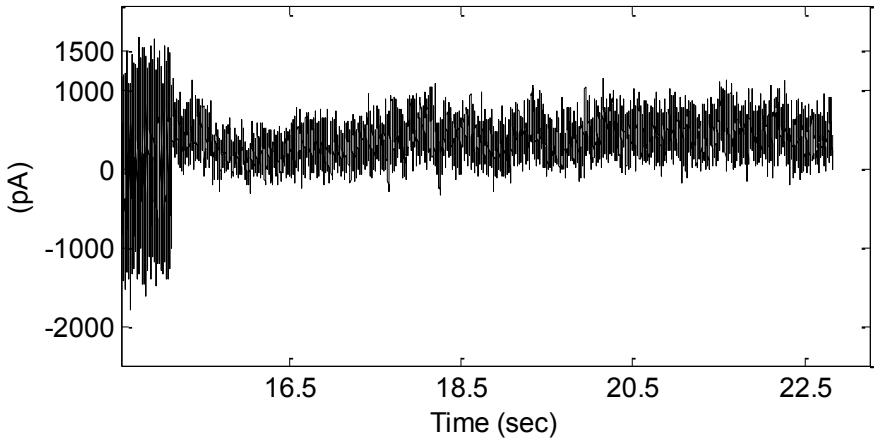
It is clear from Figures 3.9 and 3.10 that the reconstructed membrane potential of the HH model using JUKF and JEKF algorithms can predict almost all spikes (except one) of the neuron. More importantly, the sub-threshold activity of the membrane potential is nicely tracked by both methods. This highlights the capability of the proposed algorithms for the HH neuronal model to predict not only the spike timing of the neuron but also estimate its sub-threshold activities. It is worth mentioning that these activities carry important information about the synaptic inputs of the neuron.

3.9.2 Real Data

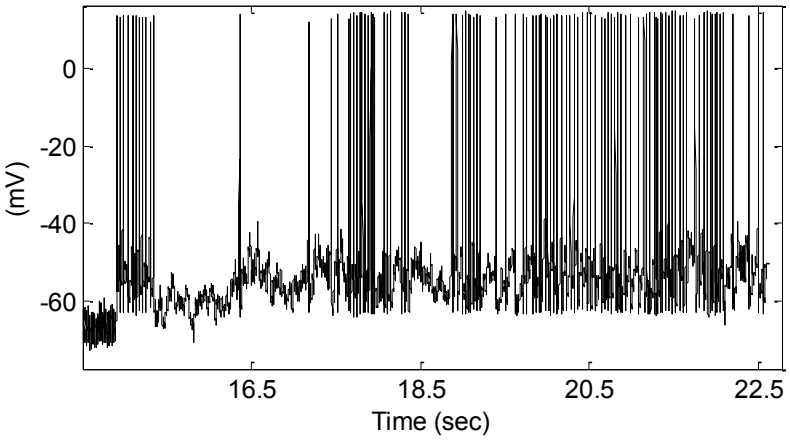
As mentioned before, the real data is provided from [84]. This data is prepared for a competition (Challenge A) supported by EPFL to verify the performance of different neuronal models for predicting the spike timing of a single neuron with regular spiking from L5 pyramidal cell of a 14-day-old Wistar rat [84]. The rat was decapitated, its brain was quickly transferred to a slicing chamber filled with iced artificial cerebrospinal fluid (ACSF), and 300 mm thick slices of the primary somatosensory neocortex were prepared for recording (see [85] and [86] for more details).

Although the neuronal models built for this data set [83] have generated good quantitative predictions of the future activity of the tested neuron under temporally structured current injection [83], it is in general difficult to compare the advantages of various models and algorithms since each model is designed for a different set of data. It is to be noted that the HH neuronal model was not used for this competition. Here, we check the feasibility of the JUKF and JEKF algorithms in estimating the entire set of the HH model parameters underlying this neuron and reconstructing its ionic dynamics. Furthermore, the estimated parameters are used to reconstruct the membrane potential. As mentioned before, our goal is not to compete with other existing algorithms that are developed only for predicting the spike timing of the neuron. Our main objective is to highlight the feasibility of the mentioned algorithms to estimate the entire parameters of the HH neuronal model from real recorded data. It is noteworthy that the hidden dynamics of the HH model that can be reconstructed using the proposed algorithms can elucidate important information about the dynamics of the ion channels of the neuron.

Figure 3.11 shows the selected part of the injected current and the corresponding recorded membrane potential used for the algorithms to be accomplished.



(a)



(b)

Figure 3.11. Injected current (a) and membrane potential (b) of a real neuron

Having the input and output of a neuron, the JUKF and JEKF algorithms are used to infer the entire parameters of the HH neuronal model including potassium, sodium and leak currents (see (3.1)). In order to provide reasonable initial values of the parameters used in

the algorithms, JUKF and JEKF are run for several segments of the data, each segment contains 200 *ms*, and then the estimated parameters are averaged over these segments. These values are then employed as the initiations for the JUKF and JEKF algorithms to estimate the entire parameters of the HH model over the whole length of the selected data shown in Figure 3.11. The initial and estimated values of each algorithm are summarized in Table 3.4. Now, one can reconstruct the membrane potential of the neuron using the estimated parameters. The reconstructed membrane potentials from the estimated parameters of JUKF and JEKF methods are shown in Figures 3.12 and 3.13, respectively.

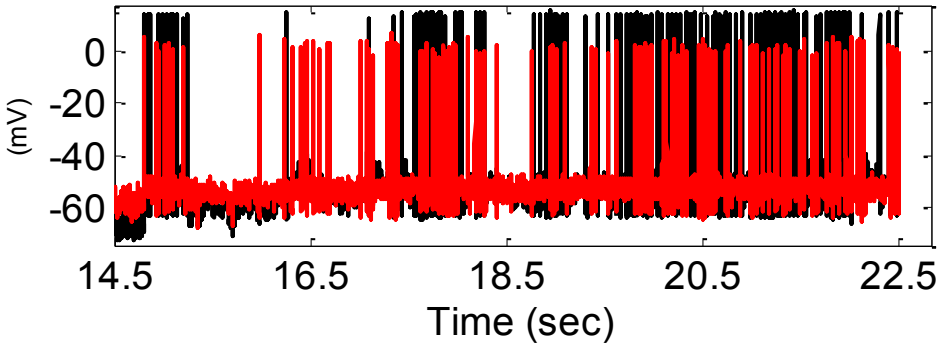


Figure 3.12. Reconstructed membrane potential of a neuron using JUKF (red dashed line) versus the recorded membrane potential (solid black line) of a real neuron.

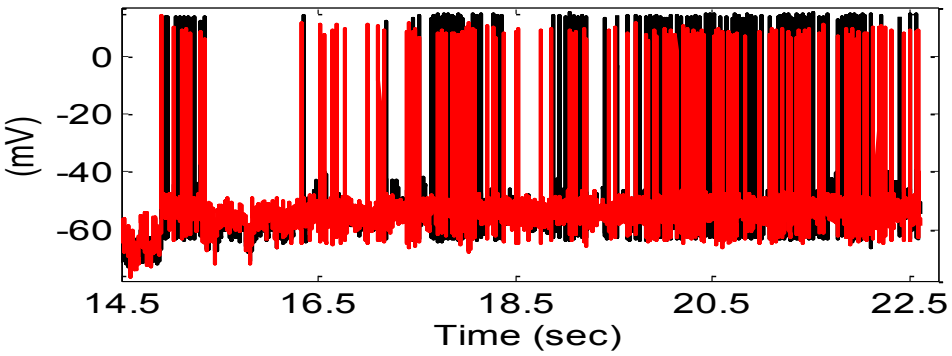


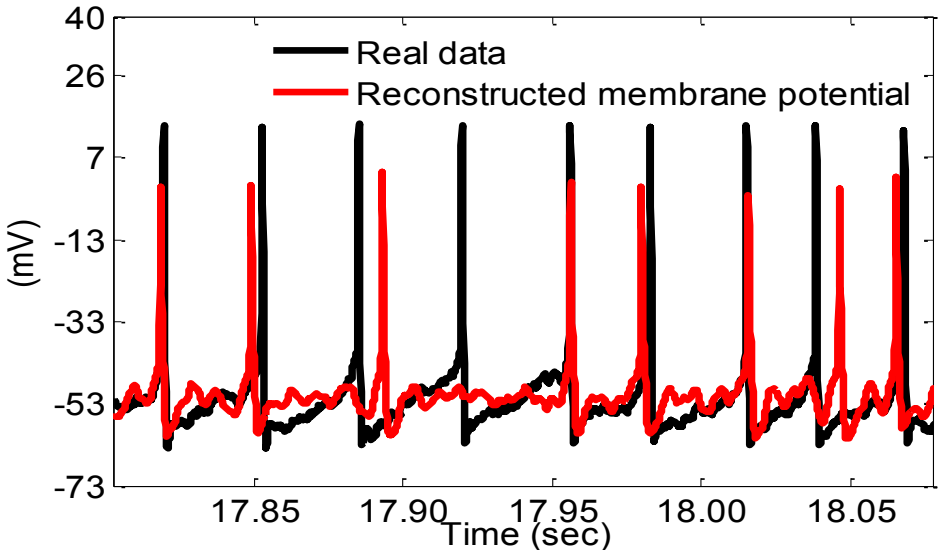
Figure 3.13. Reconstructed membrane potential of a neuron using JEKF (red dashed line) versus the recorded membrane potential (solid black line) of a real neuron.

Table 3.4. Initial values and estimated parameters of the HH neuronal model

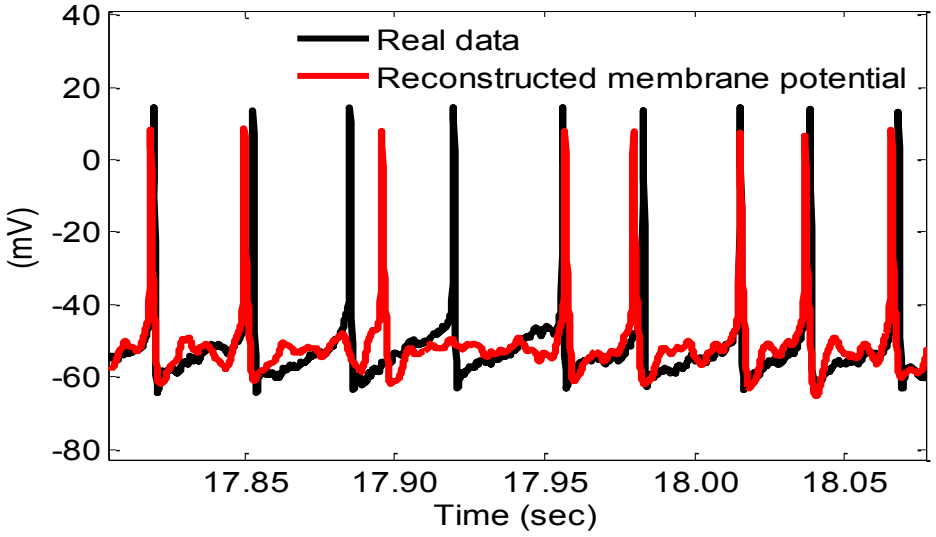
Parameter	Unit	Initial value	Estimated value	
			JEKF	JUKF
g_{Na}	mS/cm^2	14	16.65	19.72
g_K	mS/cm^2	3	5.24	2.55
g_L	mS/cm^2	0.10	0.10	0.01
E_{Na}	mV	20	24.23	23.43
E_K	mV	-66	-68.17	-65.80
E_L	mV	-48	-46.39	-45.91
$V_{th(n)}$	mV	-56	-53.34	-56.22
$V_{th(m)}$	mV	-43	-43.07	-42.10
$V_{th(h)}$	mV	-54	-53.82	-48.05
S_n	-	0.09	0.10	0.08
S_m	-	0.32	0.34	0.52
S_h	-	0.13	0.15	0.38
t_n	$Msec$	5.80	7.32	5.78
t_m	$Msec$	0.66	0.69	0.63
t_h	$Msec$	8.55	8.45	7.83

As can be seen from Figures 3.12 and 3.13, both algorithms can approximately generate all the spikes the real neuron does. In order to better observe the temporal resolution and

the shape of the reconstructed spikes, one segment of the above figures, between 17.80 *sec* – 18.10 *sec*, is zoomed in and shown in Figure 3.14.



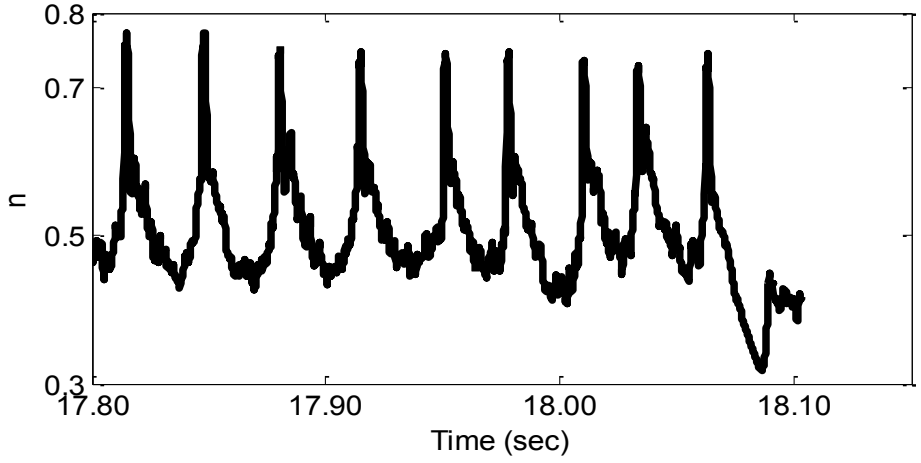
(a)



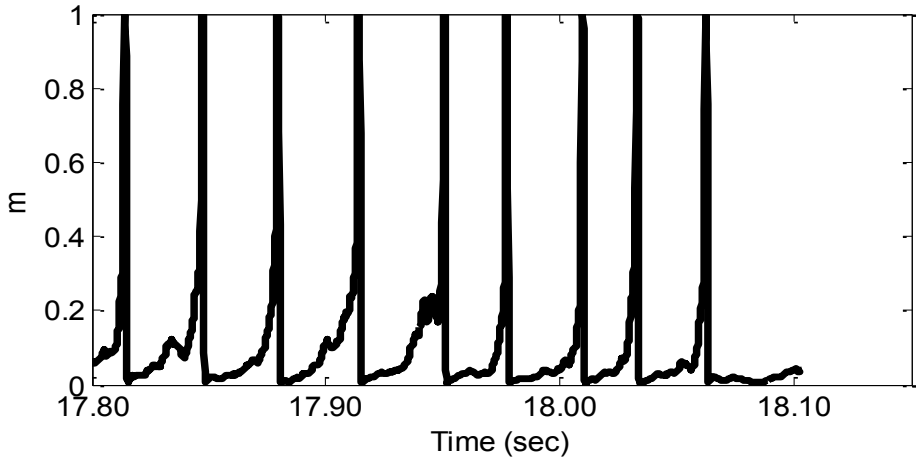
(b)

Figure 3.14. Reconstructed membrane potential form the estimated parameters of the HH neuronal model using (a) JUKF and (b) JEKF

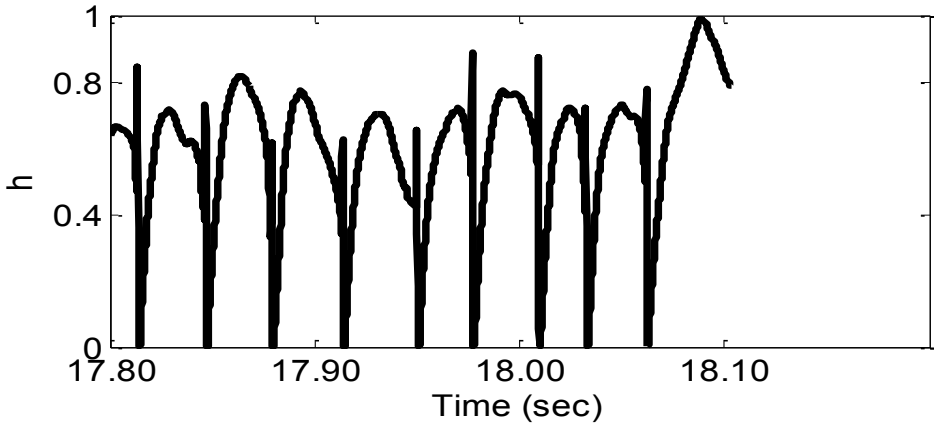
These figures demonstrate that the HH neuronal model can predict the spike timing of a real neuron under temporarily structured stimulus (the stimulus used for the recorded data). More importantly, the ion channel dynamics of the neuron, namely, the dynamics of the potassium and sodium currents, can be identified using the proposed algorithms. The corresponding hidden dynamics, namely, *n*, *m* and *h*, are reconstructed using both JUKF and JEKF algorithms and shown (for the mentioned time period) in Figures 3.15 and 3.16, respectively.



(a)

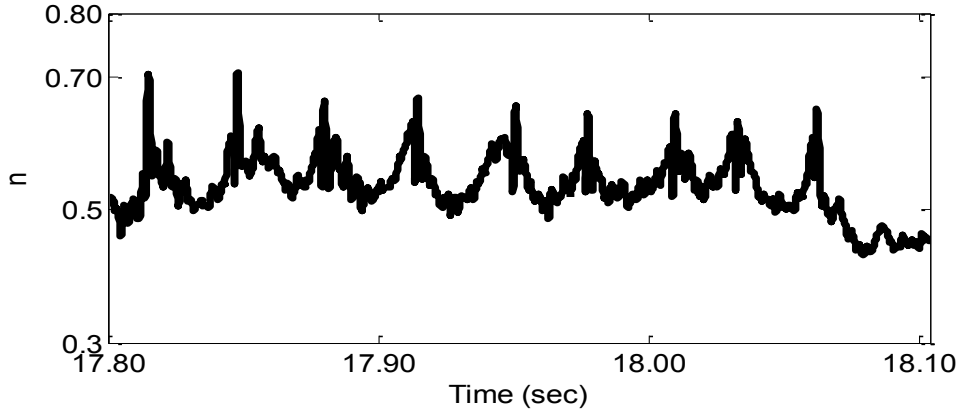


(b)

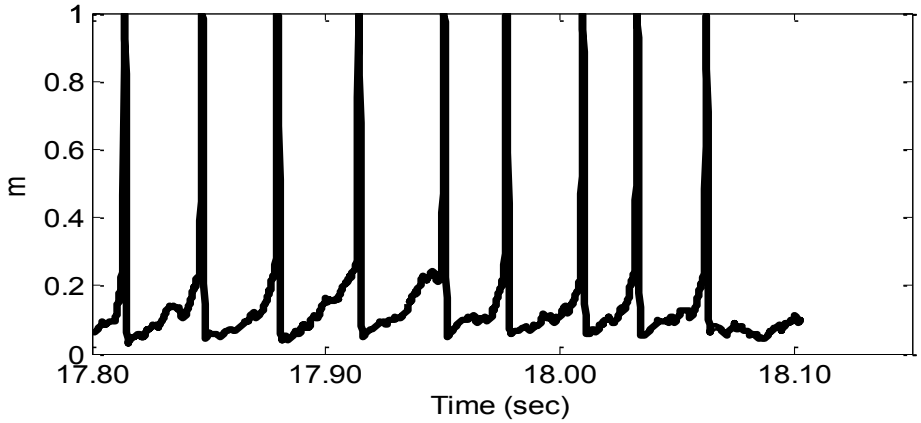


(c)

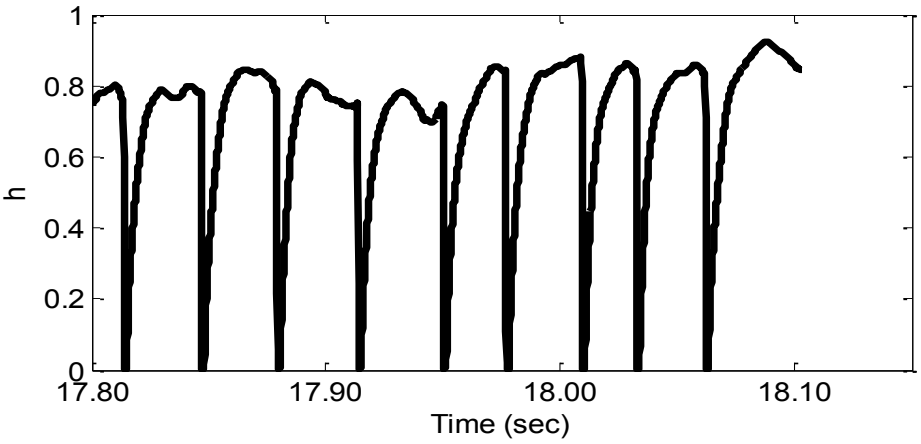
Figure 3.15. Reconstructed dynamics (a) *n*, (b) *m* and (c) *h* of a real neuron using JUKF



(a)



(b)



(c)

Figure 3.16. Reconstructed dynamics (a) *n*, (b) *m* and (c) *h* of a real neuron using JEKF

As can be observed from Figures 3.15 and 3.16, the hidden dynamics of a real neuron can be revealed by applying JUKF and JEKF algorithms to HH neuronal model. These dynamics describe how ion channels of a real neuron change in response to the injected current. This is the main advantage of the HH neuronal model in comparison with other models that only predict the spike timing of the neuron. As mentioned before, no spike prediction based on detailed biophysical neuronal model was accomplished in the mentioned competition [81]. Since our objective was verifying the feasibility of JUKF and JEKF for estimating the entire parameters of the HH neuronal model from the recorded membrane potential of a real neuron, and highlighting the capability of these algorithms in identifying the hidden dynamics of a real neuron, no quantitative comparison between our methods and existing methods in the literature has been done. Finally, it is to be noted that the existing methods in the literature only predict the spikes and cannot reveal the dynamics of the neuron.

3.10 Summary

A comprehensive study of different types of KF methods, namely, JUKF, DUKF, JEKF and DEKF, and their application to predict the hidden states and estimate the unknown parameters of the HH neuronal model, have been presented in this chapter. All the methods have been mathematically justified. Simulation results have demonstrated the high accuracy of the proposed methods (particularly JUKF, JEKF and DEKF) in the prediction and estimation of the hidden states and the unknown parameters of the HH neuronal model. In particular, the EKF-based methods exhibit a performance equivalent to that of JUKF, especially for high SNRs (> 5dB). The KF-based algorithms (as well as particle filtering) provide a possibility of trial-to-trial tracking of the dynamics of the HH model, which would be impossible using conventional methods. The results of our simulations have indicated that the performance of the EKF approaches is equivalent to that of the JUKF although the EKF-based methods, especially JEKF, are much faster and therefore are more applicable in real-time application such as dynamic clamp. Moreover, JUKF and JEKF have been developed for the HH model to estimate its entire parameters including maximum conductances, reversal potentials and ion channels kinetics. In order to verify the feasibility of these methods in estimating the entire parameters of the HH model, two experiments have been conducted. The accuracy of developed JUKF and JEKF has been confirmed in a simulation study. Moreover, the results of applying these algorithms to real data have indicated the capability of these algorithms in revealing the ion channels' dynamics of a real neuron.

Chapter 4

Inferring Excitatory and Inhibitory Synaptic Inputs

4.1 Introduction

Time-varying excitatory and inhibitory synaptic inputs govern neuronal activities and convey information in the brain. Interaction of these inputs constructs the shape of the receptive fields and can elucidate the synaptic mechanism underlying the functional activities of neurons. Therefore, inferring synaptic inputs from neuronal recordings is an important topic of interest in neuroscience [87-90] and [91]. In many cases, intercellular recordings of membrane potential (or current) under pharmacological blockade spiking activities are used to estimate synaptic inputs. Estimating synaptic inputs based on the averaging of many trials and linear regression fitting, which is commonly used, is not always the best methodology because the trial-to-trial variations of synaptic inputs are ignored. The significance of such variations in understanding the neuronal mechanisms (especially spontaneous) of the brain activity and their key roles in information

processing are well reviewed in [92]. Figure 4.1 shows the problem of inferring excitatory and inhibitory synaptic inputs from recorded membrane potential in a typical biological neural network. In this figure, each neuron is connected to its neighbor and receives excitatory and inhibitory synaptic conductances (or inputs) from other neurons (which do not belong to the mentioned network).

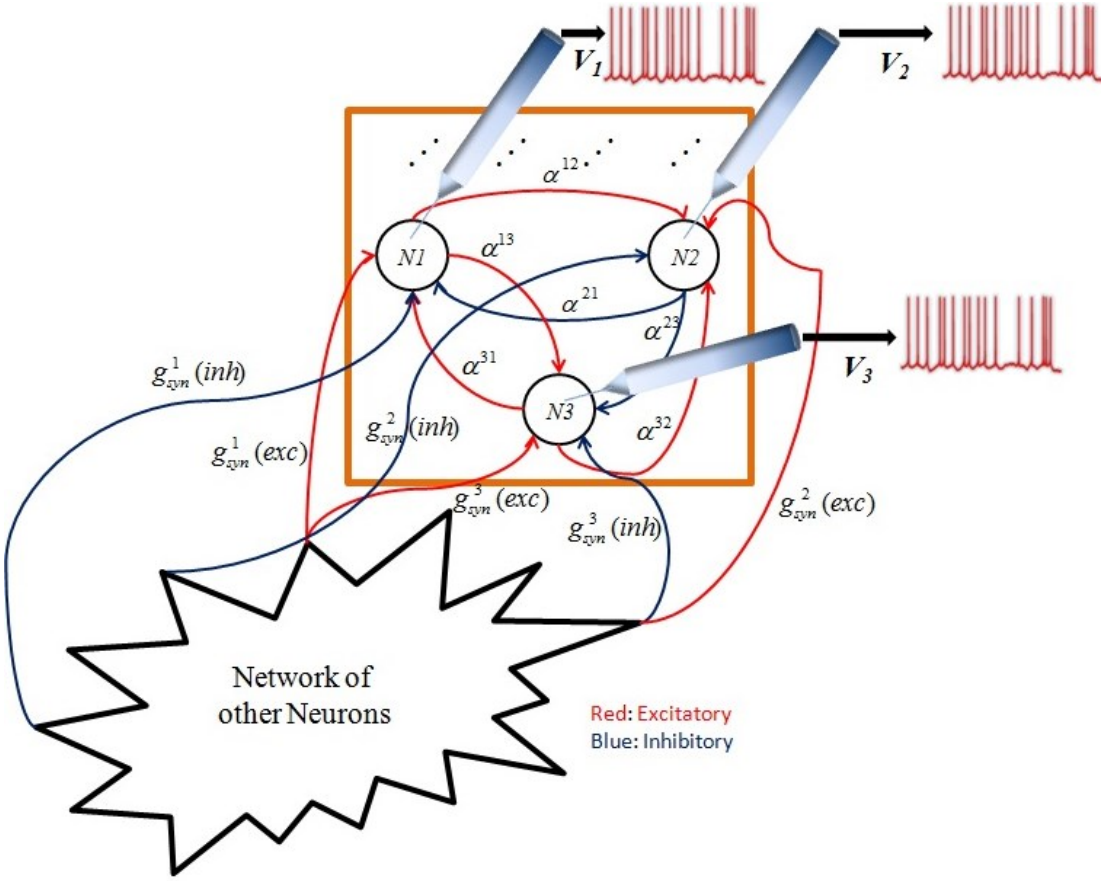


Figure 4.1. Inference of the excitatory and inhibitory synaptic inputs (or equivalently synaptic conductances) of a neuron from recorded membrane potentials in a typical biological neural network. α^{ij} represents the coupling weight between the i^{th} and the j^{th} neurons (effective from neuron i to j), $g_{syn}(exc)$ and $g_{syn}(inh)$ respectively show the excitatory and inhibitory synaptic conductances received by the i^{th} neuron.

As mentioned in Chapter 3, Bayesian-based methods provide the capability of trial-to-trial tracking the dynamics of a single neuron. Furthermore, as will be seen in Section 4.2, the excitatory and inhibitory synaptic conductances can be considered as the dynamics of a single neuron. In this regard, it is necessary to mention two recent studies [38] and [93] that have used the well-known Bayesian approach to infer the synaptic conductances from single trial of recorded membrane potential. In both studies, promising results were reported in low observation noise. Kobayashi et al. [38] considered the Ornstein-Uhlenbeck stochastic model with time-dependent mean and variance as the neuronal model. Kalman filtering (KF) was then used to track these statistical moments from recorded membrane potential. Paninski et al. [93] used a compact neuronal model associated with two differential equations representing the dynamics of the excitatory and inhibitory synaptic conductances. Then, the sequential Monte-Carlo method or particle filtering (PF) was derived for filtering/smoothing the dynamics of the model. Finally, an expectation-maximization (EM) algorithm, both in parametric and non-parametric manner, was used to infer the time-varying mean of the synaptic conductances. Since the above-mentioned studies used the Bayesian approach, the synaptic input's distributions have to be known as *a priori* knowledge. This is the major theoretical drawback of these methods, since synaptic distributions are unknown in real neurons. Moreover, Kobayashi et al [38] assumed that all excitatory or inhibitory synaptic weights are identical in order to obtain an explicit relation between the excitatory/inhibitory synaptic inputs and the mean and variance of the total input current (sum of excitatory and inhibitory inputs). However, this assumption does not necessarily hold in real neurons.

The difficulty in estimating the time course of both excitatory and inhibitory synaptic inputs from only a single trial of recorded data as compared with other conventional methods (averaging and estimating the mean of synaptic inputs) is that the problem is underdetermined, since two unknown variables have to be estimated at each time instant. In order to overcome this difficulty, we propose, in this chapter, a robust recursive algorithm, based on Gaussian mixture Kalman filtering (GMKF), for filtering/smoothing the dynamics of a compact neuronal model (including synaptic conductances) followed by an EM algorithm to infer the statistical parameters of such synaptic inputs [35, 94]. Our methodology provides more degrees of freedom for these inputs by estimating their distributions with a Gaussian mixture model (GMM). As we are dealing with Gaussian distribution for each mixand, KF is considered optimal, which is also faster and easier than the PF approach [93]. Moreover, we will prove that the least square (LS) estimation technique that is commonly used in the literature is a biased-estimator for the time-varying synaptic inputs. This drawback of the LS method may challenge our understanding on the balance of excitatory and inhibitory synaptic conductance [87].

4.2 A Neuronal Model Including Excitatory and Inhibitory Synaptic Inputs

As already pointed out in Section 2.1, despite the wide variety of synapses in the brain, they all convey messages only in two types: excitatory or inhibitory. In fact, excitatory synaptic inputs increase the probability of producing action potentials in the target neuron, while the inhibitory synaptic inputs reduce that probability. In order to consider

the influence of the excitatory and inhibitory synaptic inputs on a single neuron, a reasonable neuronal model similar to that of [93] is employed here. This neuronal model represents the dynamics of a single neuron that receives synaptic inputs from other neurons. The observation is the sub-threshold membrane voltage where the active channels are pharmacologically blocked. This model can be expressed as follows.

$$\begin{cases} V(t+1) = V(t) + dt[g_L(E_L - V(t)) + g_E(t)(E_E - V(t)) + g_I(t)(E_I - V(t))] + w(t) \\ g_E(t+1) = g_E(t) - dt\dfrac{g_E(t)}{\tau_E} + N_E(t) \\ g_I(t+1) = g_I(t) - dt\dfrac{g_I(t)}{\tau_I} + N_I(t) \end{cases} \tag{4.1}$$

where V, g_E and g_I are the dynamics of the neuron indicating the membrane potential and excitatory and inhibitory synaptic conductances, respectively, $w(t)$ is white Gaussian noise of variance σ^2_w, $N_E(t)$ and $N_I(t)$ are the instantaneous excitatory and inhibitory synaptic inputs to the neuron at time step t [6, 65, 93], respectively, and dt is the time bin that may differ from the voltage recording sampling time [93]. Note that the time index t takes integer values between 0 and T, where $T \times dt$ is the entire (physical) time of recording. We assume that these time steps are equidistant. Similar to [93] and [38], the reversal potentials E_L, E_E, and E_I, the leakage conductance g_L, and the synaptic time constants τ_E and τ_I are known.

Our objective in this chapter is to assess the time trace of the excitatory and inhibitory synaptic conductances, g_E and g_I, as well as the corresponding synaptic inputs N_E and N_I from noisy membrane potential using the known Bayesian approach. To optimally reconstruct the time course of the excitatory and inhibitory synaptic conductances, we

have to determine the probability distributions of the corresponding synaptic inputs, as the *a priori* knowledge in the Bayesian approach. Most of previous studies used Poisson distribution as the distribution of the synaptic inputs [38] (see also [93] that derives PF for the exponential distribution). Here we use a weaker assumption about the distributions of the synaptic inputs, that is, the probability distribution function of the synaptic input can be estimated by a finite number of weighted Gaussians — Gaussian mixture model (GMM). Moreover, by identifying and tracking each Gaussian component with KF, we propose a general GMKF-based algorithm. The probability distribution functions of the excitatory and inhibitory synaptic inputs are given by

$$\begin{aligned} p(N_E(t)) &= \sum_{j=1}^{G} \alpha_j N\left(\mu_{\boldsymbol{E}_j}(t), \Sigma_{\boldsymbol{E}_j}(t)\right),\ N_E(t) \geq 0 \\ p(N_I(t)) &= \sum_{j=1}^{G} \alpha_j N\left(\mu_{\boldsymbol{I}_j}(t), \Sigma_{\boldsymbol{I}_j}(t)\right),\ N_I(t) \geq 0 \end{aligned} \tag{4.2}$$

where $\mu_{\boldsymbol{E}_j}(t)$ and $\mu_{\boldsymbol{I}_j}(t)$ are, respectively, the mean of the excitatory and inhibitory inputs at time t that belong to the j^{th} mixand ($j \in \{1:G\}$). Here, G is the number of mixands. Similarly, $\Sigma_{\boldsymbol{E}_j}(t)$ and $\Sigma_{\boldsymbol{I}_j}(t)$ are the time-varying variances of these inputs at time t, and α_j is the weight for the j^{th} mixand. Our goal is to estimate $N_E(t)$ and $N_I(t)$ in (4.1) by using the GMM in (4.2). To this end, we use extended Kalman filtering (EKF) to estimate the dynamics of (4.1) followed by the well known EM algorithm to infer the statistical parameters of the synaptic inputs, $\mu_{\boldsymbol{E}_j}(t)$, $\mu_{\boldsymbol{I}_j}(t)$, $\Sigma_{\boldsymbol{E}_j}(t)$, and $\Sigma_{\boldsymbol{I}_j}(t)$ in (4.2). By using these statistics as the *a priori* knowledge, we repeat our algorithm until no considerable changes in the estimated dynamics occur.

4.3 Inferring Time-Varying Excitatory and Inhibitory Synaptic Conductances

As mentioned in Section 4.2, we would like to infer the excitatory and inhibitory synaptic conductance, or equivalently the synaptic inputs, only from noisy membrane potential. Moreover, it is previously discussed in Section 4.1 that the well-known least square (LS) method, belong to the optimization-based category, is widely used in neuroscience. This technique for inferring the excitatory and inhibitory synaptic conductances is presented in subsection 4.3.1. Moreover, it is shown there that the LS is a biased estimator and not able to track trial-to-trial variability of the synaptic inputs. However, trial-to-trial estimating these inputs from noisy membrane potential can reveal the drivers of neurons and play an important role in our understanding of information processing in neuronal circuits. In subsection 4.3.2, we present a general recursive framework for the identification of nonlinear and time-varying systems based on which our methods are proposed.

4.3.1 Least Square (LS) Method

Here we show that the LS estimation of the synaptic conductances is biased if their variations are correlated with those of the membrane potential. From (4.1) we can easily show that the dynamics of membrane potential during the blockade of sodium channels (no spikes) satisfy:

$$C_M \frac{dV}{dt} = -g_L(V - V_L) - g_E(V - V_E) - g_I(V - V_I) + I \tag{4.3}$$

where C_M is the membrane capacitance, and I is the external input. We assume that the excitatory/inhibitory synaptic conductances, g_E and g_I, are stochastic variables and denote their trial-means by $\bar{g}_E$ and $\bar{g}_I$, and trial-to-trial variations by $\Delta g_E \xi_E$ and $\Delta g_I \xi_I$, where Δg_E and Δg_I are respectively the amplitudes of variations in excitatory and inhibitory synaptic conductances, and ξ_E and ξ_I are the corresponding stochastic parts modeled by white Gaussian noise of zero mean and unit variance, i.e., the assumption that the excitatory and the inhibitory synaptic conductances are independent. We also denote the trial-mean and the trial-to-trial variation of the membrane potential by $\bar{V}$ and $\Delta V \xi_V$ (ξ_V, that is correlated with ξ_E and ξ_I, is modeled by Gaussian noise of zero mean and unit variance), respectively. With these notations the trial-mean of the membrane potential is described by:

$$C_M \frac{d\bar{V}}{dt} = -g_L(\bar{V} - V_L) - \hat{g}_E(\bar{V} - V_E) - \hat{g}_I(\bar{V} - V_I) + \bar{I} \tag{4.4}$$

where $\bar{I}$ is the trial-mean of the external input and the LS estimation of the excitatory and inhibitory conductance is given by (see Appendix B for more detail):

$$\hat{g}_E = \bar{g}_E + \frac{\Delta g_E \Delta V \overline{\xi_E \xi_V}}{\bar{V} - V_E} \quad , \quad \hat{g}_I = \bar{g}_I + \frac{\Delta g_I \Delta V \overline{\xi_I \xi_V}}{\bar{V} - V_I} \tag{4.5}$$

where $\hat{g}_E$ and $\hat{g}_I$ are the estimated excitatory and inhibitory synaptic conductances, respectively. The interpretation of (4.5) is: the LS estimation is biased if the trial-to-trial variation of the synaptic conductances and the membrane potential are correlated, which is usually the case for current-clamp recordings (recording membrane potential via clamping the injected current in different levels). This bias is negligible if the fluctuation

of the synaptic conductance or the membrane potential is small. For example, fluctuation of the membrane potential could be kept small in voltage-clamp (VC) recordings [87], although perfect control of the membrane potential along a spatially extended neuron is often difficult [91, 95]. In the special case where excitatory and inhibitory synaptic conductances are independent and changing much slower than the dynamics of the membrane potential, (4.5) is simplified as:

$$\hat{g}_E = \bar{g}_E - \frac{\Delta g_E^2}{g_L + \bar{g}_E + \bar{g}_I} \quad , \quad \hat{g}_I = \bar{g}_I - \frac{\Delta g_I^2}{g_L + \bar{g}_E + \bar{g}_I} \tag{4.6}$$

This result indicates that the LS method underestimates the excitatory and inhibitory synaptic conductance. Furthermore, if we denote by <•> the temporal-average of the function represented, the LS estimation of the covariance between excitatory and inhibitory conductance differs from the truth by:

$$\begin{aligned} &(\langle \hat{g}_E \hat{g}_I \rangle - \langle \hat{g}_E \rangle \langle \hat{g}_I \rangle) - (\langle \bar{g}_E \bar{g}_I \rangle - \langle \bar{g}_E \rangle \langle \bar{g}_I \rangle) \\ &= \left[\left\langle \frac{1}{g_L + \bar{g}_E + \bar{g}_I} \right\rangle \langle \bar{g}_I \rangle - \left\langle \frac{\bar{g}_I}{g_L + \bar{g}_E + \bar{g}_I} \right\rangle \right] \Delta g_E^2 \\ &+ \left[\left\langle \frac{1}{g_L + \bar{g}_E + \bar{g}_I} \right\rangle \langle \bar{g}_E \rangle - \left\langle \frac{\bar{g}_E}{g_L + \bar{g}_E + \bar{g}_I} \right\rangle \right] \Delta g_I^2 + O(\Delta g^4) \end{aligned} \tag{4.7}$$

Note that this difference is always positive because of the concavity of *1/x* function (see Appendix B). This means that the LS method overestimates the excitatory and inhibitory covariance in the above case if the fluctuation of the synaptic conductance is small.

4.3.2 General Recursive Framework

A general recursive framework for identification of nonlinear time-varying systems is schematically drawn in Figure 4.2. It tracks the hidden dynamics and estimates the (statistical) parameters of a dynamical system, *S*, which is defined as:

$$S:\begin{cases}\boldsymbol{x}(t+1)=F[\boldsymbol{x}(t)]+\boldsymbol{v}(t)\\ y(t+1)=H[\boldsymbol{x}(t+1)]+\varepsilon(t)\end{cases} \tag{4.8}$$

where F and H are the transition and observation functions, respectively, and $\boldsymbol{v}(t)$ and $\varepsilon(t)$ are the system noise (or the unknown stochastic inputs) and the observation noise, respectively. In Figure 4.2, $\boldsymbol{\theta}$ stands for the statistical parameters of $\boldsymbol{v}$ and ε, e.g., the mean and variances. The objective of the recursive algorithm shown in Figure 4.2 is to estimate/track the dynamics of *S* as well as infer the statistical parameters of the stochastic sources $\boldsymbol{v}$ and ε. Although this framework has been used in [65, 93], we show its effectiveness and usefulness in estimating both the hidden states of a system (in a state-space model as well as those modeled as convolution relationship) and the statistics of its input. The recursive algorithm begins with an arbitrary initiation followed by filtering/smoothing steps (2 & 3). These filtering/smoothing steps are necessary to identify the hidden dynamics of *S*. Accomplishing this step and calculating the statistics (mean, variance, etc.) of such dynamics, the parameters of the stochastic sources can be inferred by using an appropriate optimization technique, e.g., the expectation-maximization (EM) algorithm. Since these parameters construct the initial values of the next iteration, the algorithm can stop with an appropriate criterion. In the next section, we develop our proposed algorithms based on this framework.

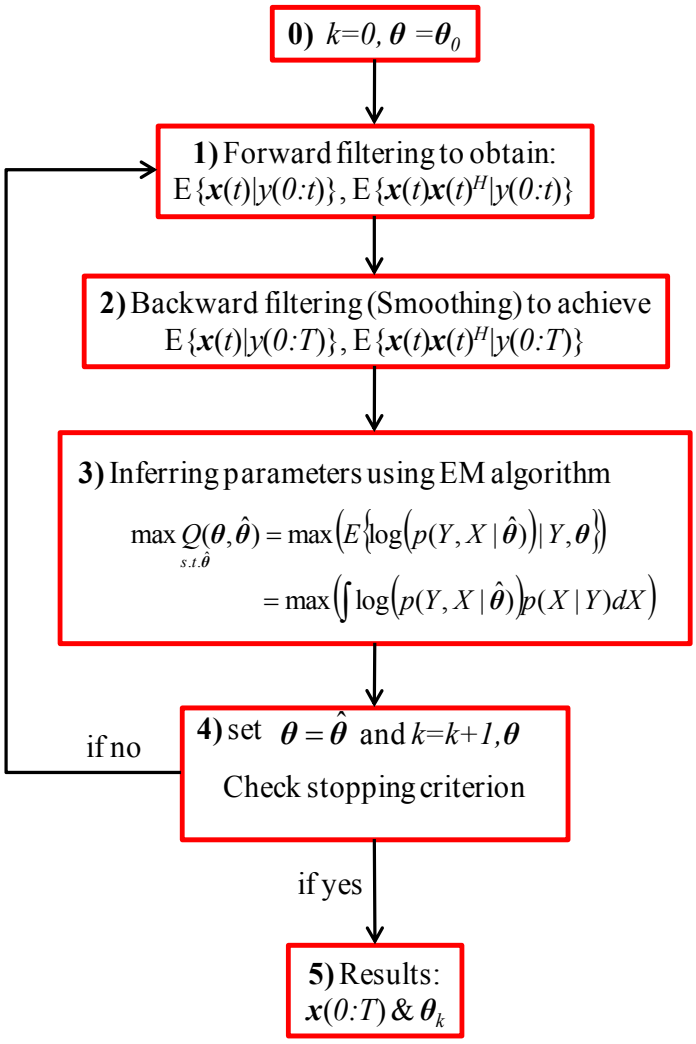


Figure 4.2. Schematic representation of the general recursive framework for tracking the dynamical states and estimating the time-varying stochastic inputs represented by system (4.7), where x is the state of the system and y is the observation. Here, k and $\boldsymbol{\theta}_0$ are the iteration number and the initial values of the statistical parameters, respectively. X and Y are abbreviations for the entire samples of $\boldsymbol{x}$ and y over time, i.e., $X=\boldsymbol{x}(0:T)$ and $Y=y(0:T)$. $\boldsymbol{\theta}$ is the unknown statistical parameters of the system noise.

4.4 Gaussian Mixture Kalman Filtering

In this section, we propose a novel method based on Gaussian mixture Kalman filtering (GMKF) that not only overcomes the aforementioned limitations of the LS method, but also gives the opportunity of trial-to-trial estimation of the excitatory and inhibitory synaptic conductance. We show that our proposed algorithm requires fewer assumptions than the recent proposals [93] and [38] that also provide trial-to-trial estimation of synaptic conductance. In particular, the proposed technique outperforms that of [93] due to its ability of estimating an unknown synaptic distribution using Gaussian mixture model (GMM).

Let $\boldsymbol{x}(t)= [V(t), g_E(t), g_I(t)]^H$ denote the vector of neuron dynamics at time t, where the superscript ^{H} represents the matrix transpose operation. The dynamical system in (4.8) can be rewritten as:

$$S:\begin{cases}\boldsymbol{x}(t+1)=F[\boldsymbol{x}(t)]+\boldsymbol{v}(t)\\ y(t+1)=C\boldsymbol{x}(t+1)+\varepsilon(t)\end{cases} \tag{4.9}$$

with

$$F[\boldsymbol{x}(t)]=\begin{bmatrix}1-dt\left(g_L+g_E(t)+g_I(t)\right), & dt\,E_E, & dt\,E_I\\ 0 & ,1-\dfrac{dt}{\tau_E}, & 0\\ 0 & ,0, & 1-\dfrac{dt}{\tau_I}\end{bmatrix}\begin{bmatrix}V(t)\\ g_E(t)\\ g_I(t)\end{bmatrix}+\begin{bmatrix}dtg_Lv_L\\ 0\\ 0\end{bmatrix} \tag{4.10}$$

Here, C is the observation vector, $[1,0,0]$, ε is the observation noise of variance σ^2_{ε}, and the distribution of the system noise (dynamical noise) $\boldsymbol{v}_t=[w(t), N_E(t),N_I(t)]^{\mathrm{H}}$ is a GMM containing G mixands.

$$p(\boldsymbol{v}_t) = \sum_{j=1}^{G} \alpha_j N\left(\boldsymbol{\mu}_{\boldsymbol{v}_j}(t), \Sigma_{\boldsymbol{v}_j}(t)\right)$$

$$\boldsymbol{\mu}_{\boldsymbol{v}_j}(t) = [0, \boldsymbol{\mu}_{\boldsymbol{E}_j}(t), \boldsymbol{\mu}_{\boldsymbol{I}_j}(t)] \ , \ \Sigma_{\boldsymbol{v}_j}(t) = \begin{bmatrix} \sigma_w^2, & 0, & 0 \\ 0, \Sigma_{E_j}(t), & 0 \\ 0, 0, \Sigma_{I_j}(t) \end{bmatrix} \tag{4.11}$$

where α_j is the probability of selecting the j^{th} mixand, and N_E and N_I, which are of our interest, describe excitatory and inhibitory synaptic inputs, respectively. Since the distribution of the system noise is a mixture of Gaussians, one may simply use KF for each mixand (recall that KF is an optimal filter when the system noise is Gaussian). The major drawback of this approach is that the number of Kalman filters required to estimate the conditional probability $p(\boldsymbol{x}(t)|y(0:t))$ increases exponentially with time [96]; therefore, the computational cost of this approach becomes very heavy. However, to eliminate this drawback, we use a parallel dynamic state space and resampling approach [96]. The aim of this approach is to keep a constant number of Kalman filters for estimating the conditional probability $p(\boldsymbol{x}(t)|y(0:t))$ upon the arrival of a new observation at t. In this regard, the conditional probability $p(\boldsymbol{x}(t)|y(0:t))$ is approximated by K filters. Then, it is obvious that $K\times G$ Kalman filters are required to represent $p(\boldsymbol{x}(t+1)|y(0:t+1))$ (see Appendix C for more details). Using a resampling technique, K filters are again selected to approximate the later probability; hence, the number of filters remains constant at the

arrival of each new observation. Consistent with this description, $p(\boldsymbol{x}(t)|y(0{:}t))$ can be expressed as the combination of K parallel Kalman filters, as given below.

$$p(\boldsymbol{x}(t) \mid y(0:t)) = \sum_{i=1}^{K} \beta_i(t)\, p(\boldsymbol{x}(t) \mid y(0:t), i) \tag{4.12}$$

where $p(\boldsymbol{x}(t)|y(0{:}t),i)$ indicates the conditional probability distribution function (pdf) of the i^{th} filter and $\beta_i(t)$ is the normalized weight corresponding to the i^{th} Kalman filter for the new observation at t. For the new observation at time instant $t+1$, the conditional pdf $p(\boldsymbol{x}(t+1)|y(0{:}t+1))$ is given by the $K\times G$ parallel Kalman filter (since $p(\boldsymbol{x}(t+1)|\boldsymbol{x}(t))$ is represented by G mixands).

$$p(\boldsymbol{x}(t+1) \mid y(0:t+1)) \approx \sum_{i=1}^{K} \sum_{j=1}^{G} \gamma_{i,j}(t+1)\, p(\boldsymbol{x}(t+1) \mid y(0:t+1), i, j) \tag{4.13}$$

where $\gamma_{i,j}(t+1)$ is the conditional probability of selecting the i^{th} filter and j^{th} mixand at the arrival of $y(t+1)$, i.e., $\gamma_{i,j}(t+1) = p(i,j|y(0{:}t+1))$. As mentioned above, to avoid increasing the number of filters at each new time, we resample to select the most K probable filters from the $K\times G$ filters used in (4.13). Consequently, (4.13) can be rewritten as

$$p(\boldsymbol{x}(t+1) \mid y(0:t+1)) \approx \sum_{i=1}^{K} \beta_i(t+1)\, p(\boldsymbol{x}(t+1) \mid y(0:t+1), i) \tag{4.14}$$

where $\beta_i(t+1)$ is obtained by selecting the K most significant values of $\gamma_{i,j}(t+1)$. In the next subsection, the KF is derived for each $i \in \{1:K\}$ and $j \in \{1:G\}$. The final estimation of the states is the combination of the results of these filters.

4.4.1 Forward Kalman Filtering

As mentioned in Section 3.3, in KF, we use a set of mathematical equations underlying the process model to estimate the current state of a system and then correct it using available sensor measurements [67]. In EKF, a truncated first-order Taylor linearization of the nonlinear process and measurement model is used to derive the underlying prediction-correction mechanism. Using (4.1), *a priori* (predicted) state estimate and error covariance matrix can be calculated at each t. Moreover, following the standard KF for linear time invariant systems, the correction step calculates *a posteriori* state estimate and error covariance matrix for this time instant. These variables will be used in the KF recursive framework for the next time instant $t+1$, regarding the arrival of a new observation. According to the above discussions, after combing results from K Kalman filters and G mixands at t, we run $K \times G$ parallel Kalman filters. Then, resampling to select K filters is accomplished before the arrival of new observation at $t+1$. For each i belonging to $\{1:K\}$ and j belonging to $\{1:G\}$, we aim to calculate the state estimate $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0:t)\}$ and state correlation matrix $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)\ (\boldsymbol{x}_{i,j}(t))^H\ |y(0:t)\}$ in the forward filtering step (see Figure 4.2) and $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0:T)\}$ and $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)\ (\boldsymbol{x}_{i,j}(t))^H\ |y(0:T)\}$ in the backward filtering (smoothing) step using KF approach, where $\mathrm{E}\{.\}$ stands for the expected value and $\boldsymbol{x}_{i,j}$ is the state vector belong to the i^{th} filter and j^{th} Gaussian mixand. For the forward filtering step, for each i and j, we can apply the EKF approach [69] as explained in Appendix D.

4.4.2 Backward Kalman filtering (Smoothing)

In this step, we obtain the smoothed state estimate $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0:T)\}$ and state correlation matrix $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)\ (\boldsymbol{x}_{i,j}(t))^H\ |y(0:T)\}$ and the corresponding weights $\gamma_{i,j}(t)$ for all $i \in \{1:K\}, j \in \{1:G\}, t \in \{0:T\}$. This step is explained in details in Appendix E. Calculating $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0:T)\}$ and $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)\ (\boldsymbol{x}_{i,j}(t))^H\ |y(0:T)\}$ in the backward filtering (smoothing) step, we can infer the statistical parameters of the system noise $\boldsymbol{v}$ via the EM algorithm.

4.4.3 Inferring Statistical Parameters via Expectation-Maximization

The EM algorithm is a robust optimization technique for inferring the parameters of models involving unobserved data [41], e.g., the excitatory/inhibitory synaptic inputs $N_E(t)$ and $N_I(t)$ in this chapter. This algorithm is guaranteed to increase the likelihood of the model at each iteration and therefore, can find a local optimum of the likelihood [93]. In this section, the EM algorithm is used to infer the statistical parameters of (4.2), i.e., the time varying mean ($\boldsymbol{\mu}_{v_j}(t)$) and the variance of the states ($\sigma^2_w, \Sigma_{v_j}(t)$), and the variance of the observation noise (σ^2_ε). Before using the EM algorithm, let us summarize what we have estimated by the mixture of Kalman filters as follows: the (smoothed) state estimate $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0:T)\}$ and state correlation matrix $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)\ (\boldsymbol{x}_{i,j}(t))^H\ |y(0:T)\}$ and the corresponding weights $\gamma_{i,j}(t)$ for all $i \in \{1:K\}, j \in \{1:G\}, t \in \{0:T\}$. Having these values we can easily calculate the final state estimate $\mathrm{E}\{\boldsymbol{x}(t)|y(0:T)\}$ as the combination of the mixtures and parallel filters.

$$E\{\boldsymbol{x}(t) \mid y(0:T)\} = \hat{\boldsymbol{x}}(t) = \sum_i \sum_j \gamma_{i,j}(t)\hat{\boldsymbol{x}}_{i,j}(t) \tag{4.15}$$

where $\hat{x}(t) = \left[\hat{V}(t), \hat{g}_E(t), \hat{g}_I(t)\right]^H$ is the state vector estimated by KF. Note that for the sake of simplicity of expressing notations, we denote $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0{:}T)\}$ by $\hat{\boldsymbol{x}}_{i,j}(t)$. To use the EM algorithm, it is essential to write the joint distribution of the states and observation, over time, as follows (X and Y denotes the entire samples of $\boldsymbol{x}$ and y over time, respectively):

$$\begin{aligned} &p(Y,X,i,j\,|\,\theta) = p(Y\,|\,X,\theta,i,j)p(X\,|\,\theta,i,j)p(i\,|\,\theta)p(j\,|\,\theta) \\ &\log p(Y,X,i,j\,|\,\theta) = \log p(i\,|\,\theta) + \log p(j\,|\,\theta) + \log p(Y\,|\,X,\theta,i,j) + \log p(X\,|\,\theta,i,j) \end{aligned} \tag{4.16}$$

We want to maximize the log of the joint probability of the states and observation via the EM algorithm for each mixture as follows.

$$\begin{aligned} \max_{s.t.\hat{\theta}} Q(\theta,\hat{\theta}) &= \max\left(E\left\{\log\left(p(Y,X,i,j\,|\,\hat{\theta})\right)\middle|\, Y,\theta\right\}\right) \\ &= \max\left(\int \log\left(p(Y,X,i,j\,|\,\hat{\theta})\right)p(X\,|\,Y,\theta)dX\right) \end{aligned} \tag{4.17}$$

where

$$p(X\,|\,Y) = \sum_{i=1}^{K}\sum_{j=1}^{G} p(i,j\,|\,Y)\,p(X\,|\,i,j,Y) \tag{4.18}$$

By doing the corresponding calculations to solve (4.18) (as described in Appendix F), we can obtain the mean and variance of each mixand (for both excitatory and inhibitory inputs). By combining them, the total mean and variance of the synaptic inputs as well as the observation noise variance are calculated. As a result, we can update the statistical parameters of the excitatory and inhibitory synaptic inputs as well as the variance of the observation noise in the M-step (see Appendix F for full derivations). Inferring all the parameters, we can initialize the next iteration of the recursive algorithm (see Fig 4.2).

The algorithm continues until no considerable changes in two consecutive iterations occur.

4.4.4 Special Case: Kalman Filtering

The simplest case of our GMKF-based algorithm is a simple Kalman filter ($G=1$ & $K=1$) for the filtering/smoothing step. By providing the sufficient statistics in these steps, the non-parametric EM algorithm gives the smoothed mean and variance (both are time-varying) of excitatory and inhibitory synaptic inputs. As a brief description of this algorithm, the probability distribution function $p(\boldsymbol{x}(t)|y(0:t))$ is approximated by only one Gaussian distribution. Therefore, $\mathrm{E}\{\boldsymbol{x}(t)|y(0:T)\}$ (or $\hat{\boldsymbol{x}}(t)$, which is given as a combination of $K\times G$ parallel filters in the GMKF) can be calculated through the standard KF. This strategy not only reduces the complexity of the GMKF-based algorithm but also results in a highly accurate reconstruction of the excitatory and inhibitory synaptic conductances in many cases. Two major issues arise from the specific choice of G=1 and K=1 that we need to clarify. First, the synaptic conductances have to be constrained as positive values. Second, the EM algorithm has to be derived based on truncated Gaussian distributions for the synaptic inputs. Note that these would not be an issue if $G > 1$ is used, since the probability of having negative synaptic conductances naturally decreases with the number of Gaussian mixiands.

The first issue can be easily addressed by using the constrained KF [97]. We use convex optimization toolbox CVX [98] to penalize the Kalman gain as follows (SDPT3 is another MATLAB package for semi-definite problem optimization that can be used):

$$K^C(t) = \arg\min \left\{ trace \left[(I - K(t)C) \sum_x^{t-1}(t)(I - K(t)C)^H + K(t)\sigma_\varepsilon^2 K(t)^H \right] \right\} \quad s.t.\ D(\boldsymbol{x}^{t-1}(t) + K(t)e(t)) \geq \boldsymbol{0}_{3\times 1} \tag{4.19}$$

where $K^C(t)$ is the constrained Kalman gain at t, $x^{t-1}(t)$ and $\Sigma^{t-1}{}_x(t)$ are the predicted state estimate and state correlation matrix at t, respectively, and D is a diagonal matrix with the values [*-1, 1, 1*] preserving the negativity of membrane potential as well as the positivity of the synaptic conductances. According to this constrained optimization, the Kalman gain, at each time t, is calculated such that the positivity of synaptic conductances is satisfied. It is noteworthy that our results show that a simple constraint on the (updated) state estimate $Dx^t(t) \geqq 0$ ($x^t(t) = x^{t-1}(t) + K(t)e(t)$ in standard KF), without applying the constrained optimization for calculating the new Kalman gain, $K^C(t)$, exhibits a performance very similar to that obtained by using (4.19). This means that the simple and conventional KF with ignoring negative synaptic conductances (zero forcing the updated $x^t(t)$ for negative synaptic conductances), can be an effective alternative for (4.19).

The second issue makes the M-step of our EM algorithm more complicated than we have presented for the GMKF-based algorithm. Here again, we have heuristically found that the standard EM algorithm assuming Gaussian distributions of the synaptic conductances works very well because the estimated synaptic conductances rarely take negative values even if the largest noise level explored in this paper is applied.

4.5 Performance of the Proposed Algorithms

We consider two different cases for our simulations to study the performance of our proposed algorithms. In the first case, we would like to highlight the capability of Bayesian approaches in comparison with the LS method in the estimation of trial-to-trial synaptic conductances. In the second Case, we aim to compare the performance of the proposed KF- and GMKF-based algorithms with the particle filtering (PF) [93] in two different conditions, namely, large and small signal-to-noise ratios (SNRs). In all the simulations, our recursive algorithm (for both KF- and GMKF-based) ran for 10 iterations, which is consistent with the previously used parameters in [93] and gives a fair condition to compare our proposed algorithms and the PF-based algorithm [93]. Other model parameters used are similar to that in [93] and summarized in Table 4.1.

Table 4.1. Characteristics of the neuron model

E_E	10 mV
E_I	-75 mV
E_L	-60 mV
g_L	80 S
τ_E	3 ms
τ_I	10 ms

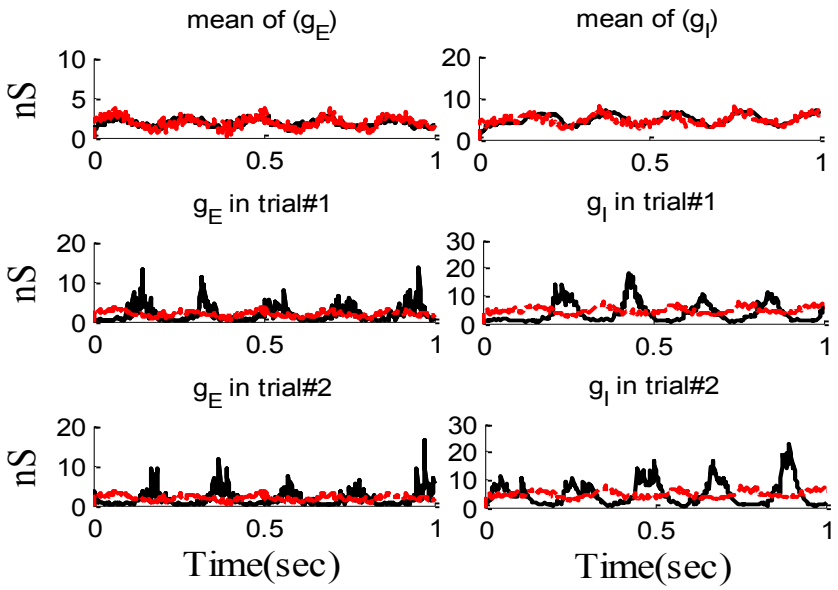
4.5.1 Bayesian Approach vs. LS method

In order to compare the performance of the LS methods and that of the Bayesian approaches, the LS method in both voltage-clamp and current-clamp modes, particle filtering (PF) [93] and our proposed GMKF algorithm are applied to two conditions,

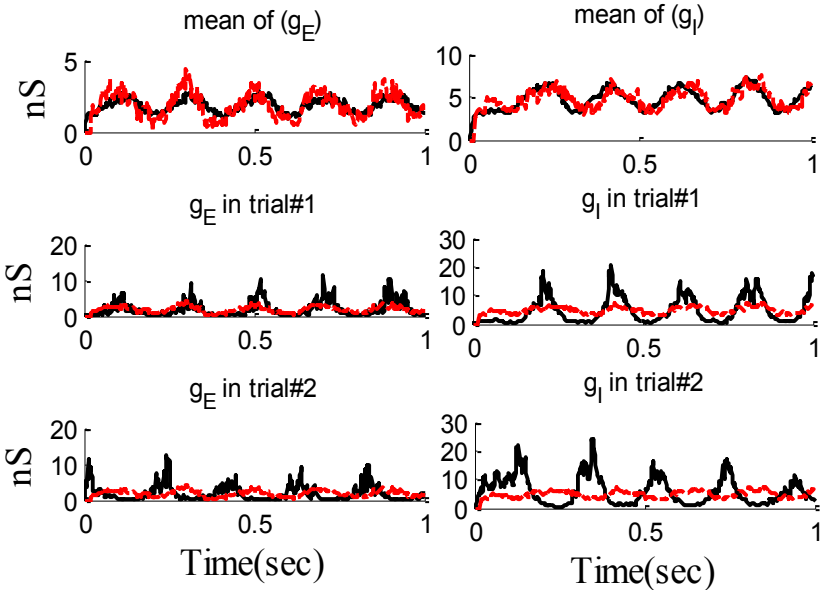
where the synaptic inputs are distributed from exponential distributions with time-varying trial-mean. In the first condition, the trial-mean is sinusoidally modulated in time with frequency of 5 Hz (the inhibition is delayed by 0.1 sec to excitation). In the second condition, excitatory and inhibitory synaptic inputs are generated from Ornestein-Uhlenbeck (O-U) process (excitation and inhibition are independent).

Example 4.1. As mentioned above, in this experiment we consider two different types of excitatory and inhibitory synaptic inputs, where both are distributed from exponential distributions with time-varying trial-mean. Firstly, the trial-mean is sinusoidally modulated in time with frequency of 5 Hz. Note that there are two sources of variability here: one is that the phase of the time-varying mean for excitation is randomly drawn in each trial from a zero mean Gaussian distribution of variance 0.25 (sec), and the other is that the actual synaptic input at each time is randomly drawn from the exponential distribution. Synthetic traces (10 trials, each lasted 1 sec) of membrane potential (in the current clamp-–mode) and somatic current (in the voltage-clamp mode with the offline series resistance compensation [87]) are recorded with three levels of the injected currents ($[-0.01, 0.01, 0.03]^{nA}$) and five levels of holding potentials ($[-10, -30, -60, -70, -80]^{mV}$), respectively, and provided for the LS method.

LS Current-clamp



LS Voltage-clamp



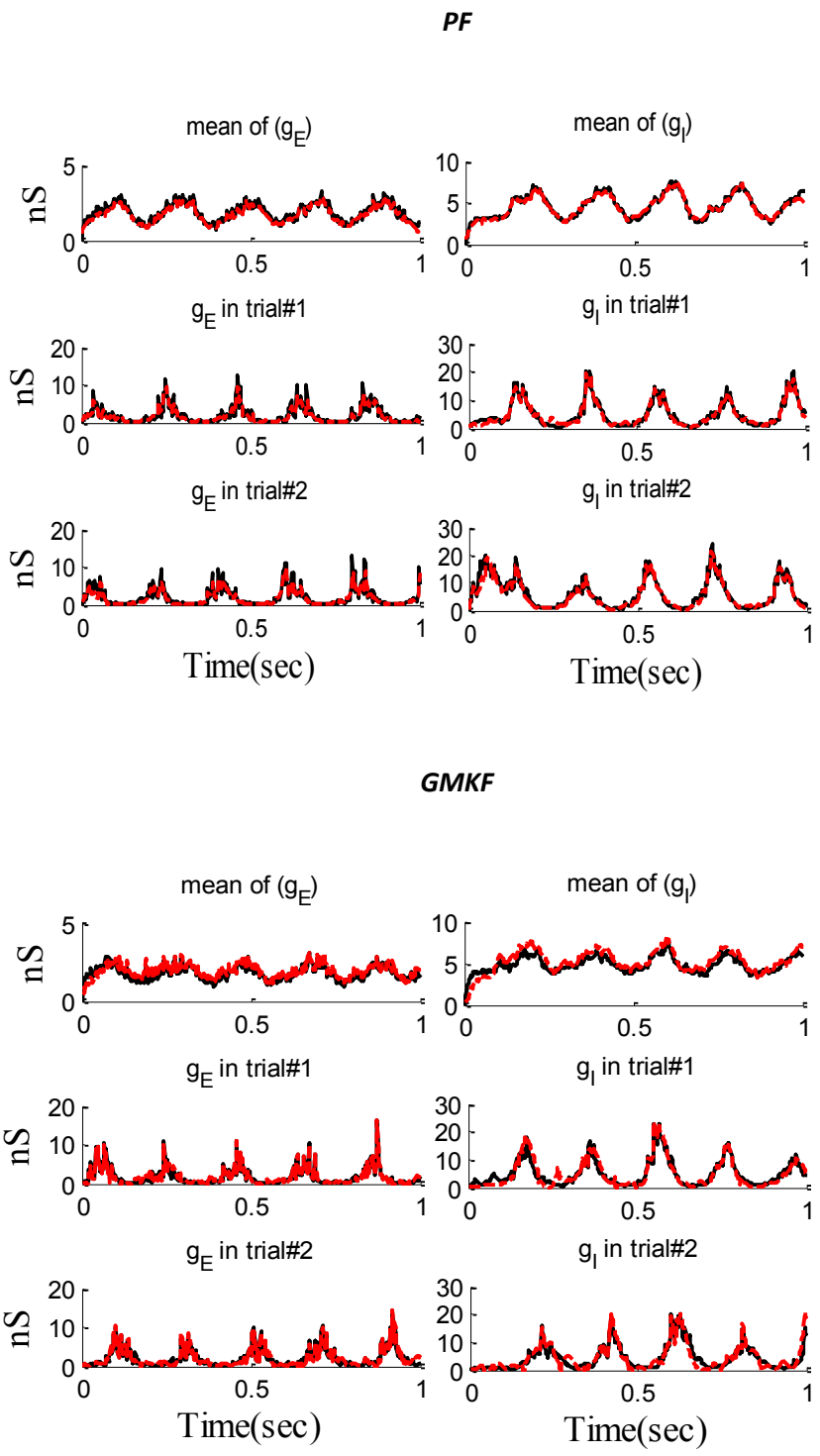


Figure 4.3. Results of Example 4.1, true values (black solid line) and estimated ones (red dashed lines).

It is noteworthy that for PF [93] and GMKF, no injected current or holding potential is needed. Figure 4.3 shows the performance of each method in estimating the excitatory and inhibitory synaptic conductance of Example 4.1.

As can be seen from Figure 4.3, both voltage-clamp and current-clamp techniques suffer from inaccurate estimations of synaptic conductances in single trials (two trials randomly selected and shown). This is in contrast to particle filtering and GMKF, which can robustly estimate the trial-mean as well as the trial-to-trial fluctuation of the synaptic conductances. Secondly, the excitatory and inhibitory synaptic inputs are generated from Ornestein-Uhlenbeck (O-U) process (excitation and inhibition are independent). We summarize the result of this condition in Table 4.2 to further quantify the accuracy of each method. In accordance with the theoretical drawbacks of the LS method, our results indicate that LS method can only estimate the trial mean; this is the reason why it does not have any variance over trials (see Table 4.2). Moreover, the LS method overestimates the correlation between excitatory and inhibitory synaptic conductance. However, GMKF and PF can properly estimate synaptic conductance at each trial and they do not suffer from overestimating the excitatory and inhibitory correlation.

Table 4.2: Excitatory/inhibitory correlation coefficients in Example 4.1 (*mean* ± *std* is shown).

True and Estimated synaptic conductances	***Correlation coefficient of trial-to-trial (g_E & g_I)***
True g_E & g_I	0.05±0.07
Estimated g_E & g_I by LS from current clamp	0.67
Estimated g_E & g_I by LS from voltage clamp	0.59
Estimated g_E & g_I by particle filtering (PF) [93]	0.12±0.14
Estimated g_E & g_I by GMKF	0.11±0.11

4.5.2 Proposed Algorithms vs. Particle Filtering [93]

In this subsection, first, we conducted two numerical experiments to demonstrate the performance of the KF-based (see Example 4.2) and GMKF-based (see Example 4.3) algorithms with large SNR, similar to [93], where the variances of system noise (σ^2_w) and observation noise (σ^2_ε) were sufficiently small. Note that estimating excitatory and inhibitory synaptic inputs in this condition was relatively easy and the results did not depend much on the algorithms used. Then, the robustness of the KF- and GMKF-based algorithms were verified in three subsequent experiments (see Examples 4.4 – 4.6) with small SNR, in which the PF-based algorithm [93] did not perform well. Time step for our simulations was 2 ms.

Example 4.2. In this experiment, the mean of the synaptic excitatory and inhibitory inputs are nonlinear functions of their synaptic fields.

$$E(N_E(t)) = \exp(k_E \xi_E(t))$$
$$E(N_I(t)) = \exp(k_I \xi_I(t))$$

where ξ_E and ξ_I were sinusoidally modulated (5 Hz) input signals and k_E and k_I were constant weights. ξ_I had 5 ms delay relative to ξ_E. The synaptic inputs, both excitatory and inhibitory, are generated from a Poisson distribution. The variance of (voltage) system noise (σ^2_w) is negligible and that of observation noise (σ^2_ε) is 0.5 mV. Obviously, since we use a non-parametric EM algorithm, ξ_E and ξ_I are unknown. Figure 4.4 indicates the results of the KF-based algorithm in estimating the excitatory and inhibitory synaptic conductances.

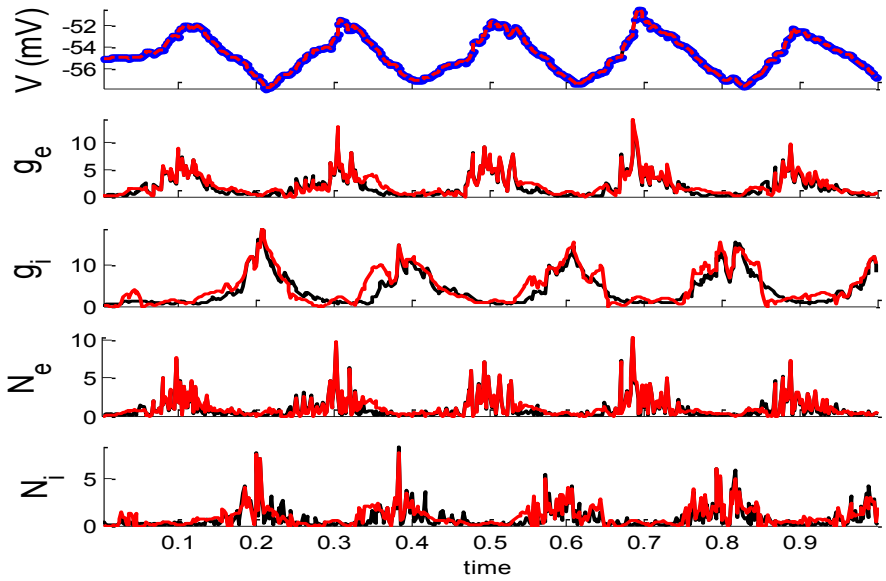


Figure 4.4. Estimating synaptic conductances and inputs given a single voltage trace of Example 4.2 using the KF-based algorithm: membrane potential (top), excitatory and inhibitory synaptic conductance (second and third from top) and excitatory and inhibitory synaptic inputs (fourth and fifth from top). Black solid lines represent true values and the red dashed lines represent the estimated ones. The blue dots at the top figure represent the observed membrane potential. The initial values of the KF-based algorithm were set as follows: The time-varying means (for both excitatory and inhibitory) were generated from a uniform distribution and their variances (for both excitatory and inhibitory) were 1 (for all times).

Example 4.3. In this experiment, the synaptic mean functions were modeled by the absolute value of random realizations of Ornstein-Uhlenbeck processes. The synaptic inputs, both excitatory and inhibitory, were generated from the Poisson distribution and the observation noise was negligible. For the GMKF-based algorithm, we set $G=2$ (number of mixands) and $K=4$ (number of Kalman filters; see our discussion about GMKF setting). The variance of the system noise (σ^2_w) was negligible and that of the observation noise (σ^2_ε) was 0.5 mV. Figure 4.5 shows the results of the GMKF-based algorithm for this experiment.

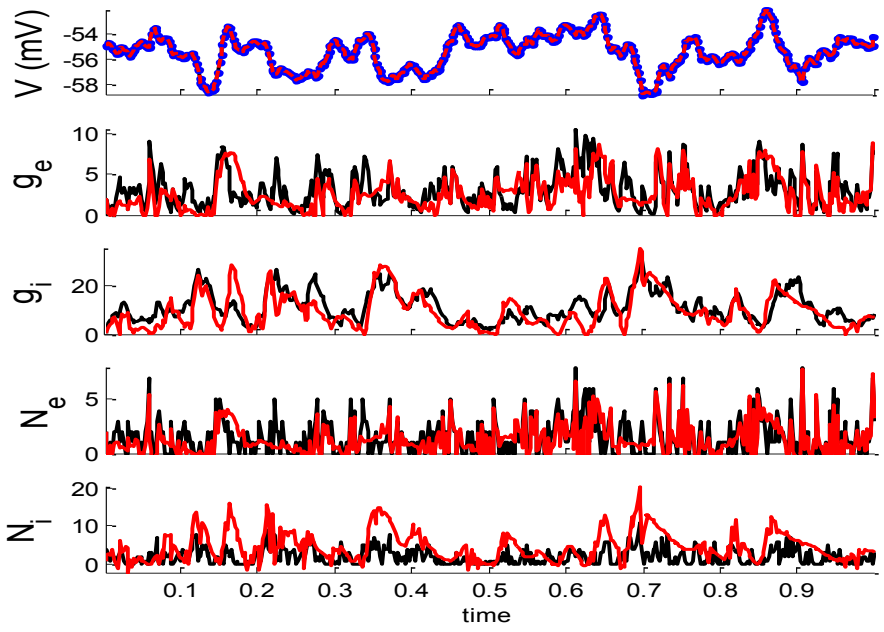


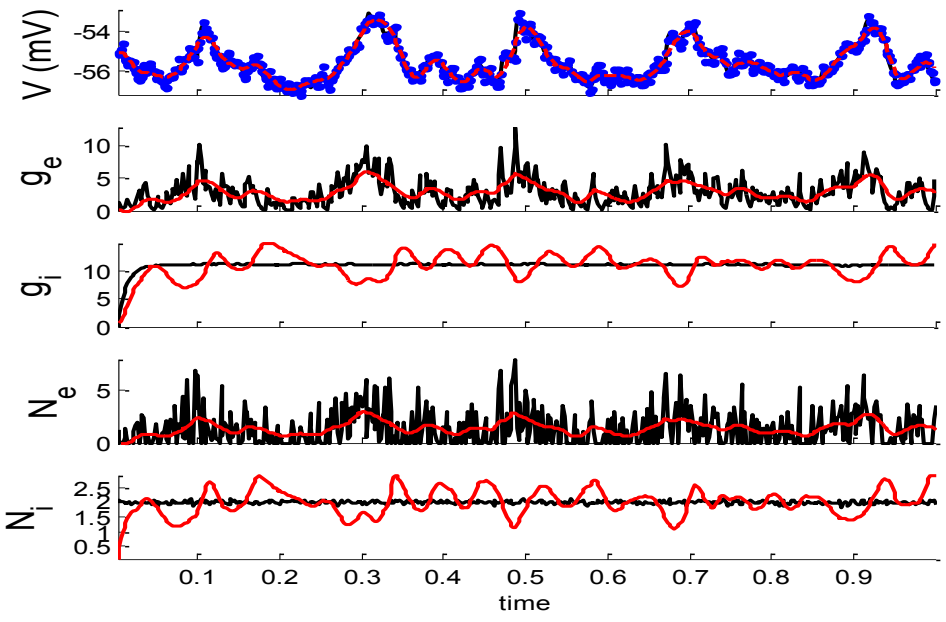
Figure 4.5. Estimating excitatory and inhibitory synaptic conductances given a single membrane potential trace of Example 4.3 using the GMKF-based algorithm. Other descriptions concerning this figure are the same as those in Figure 4.4. The initial values of the GMKF-based algorithm ($G=2$, $K=4$) were set as follows: the time-varying means (for both excitatory and inhibitory) were generated from a uniform distribution and their variances (for both excitatory and inhibitory) were 1 for both mixands (for all times).

As can be seen from Figures 4.4 and 4.5, both the KF- and GMKF-based algorithms accurately identify the excitatory and inhibitory synaptic inputs. These results are not very surprising given the large SNR used in these experiments. In fact, the PF-based algorithm could also accurately estimate synaptic inputs under similar conditions [87]. In the following experiments, we explore cases with a small SNR.

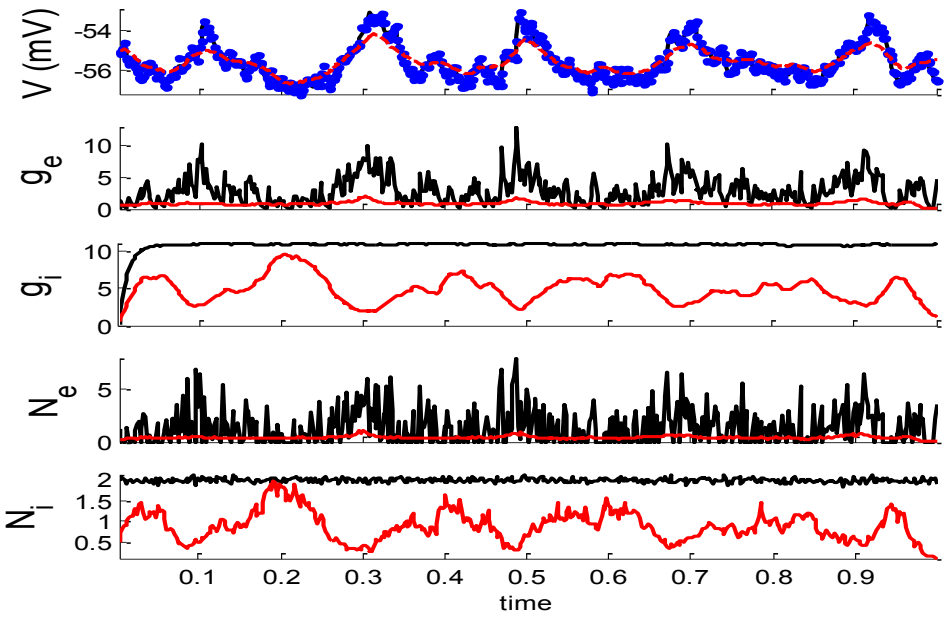
Example 4.4. In this experiment, the mean of the synaptic input of excitatory was a cosine function (amplitude 2 and frequency of 5Hz) and that for the inhibitory was a constant value (time-independent). Then, the synaptic inputs were generated from a Gaussian distribution of variance 1.5 and 0.05, respectively, for excitatory and inhibitory

inputs. The small variance of the inhibitory synaptic input generated a very narrow distribution function (almost delta function). The variances of the membrane voltage (σ^2_w) and observation noise (σ^2_ε) were 10^{-2} mV and 3 mV, respectively. These parameters were chosen not because they are physiologically realistic, but they illustrate differences in the algorithms. Figure 4.6 shows the results of the KF- and PF-based algorithms in estimating the synaptic conductances from the observed noisy membrane potential generated in this experiment.

As can be seen from Figure 4.6, g_E and g_I as well as the membrane voltage are better estimated using the KF-based algorithm. It is clear that the PF-based algorithm could not track either g_E or g_I. Figure 4.7 shows the distributions of excitatory and inhibitory synaptic conductances. It shows that the KF-based algorithm could estimate the true distributions of g_E and g_I very well, while the PF-based algorithm failes, especially for the inhibitory synaptic conductance.

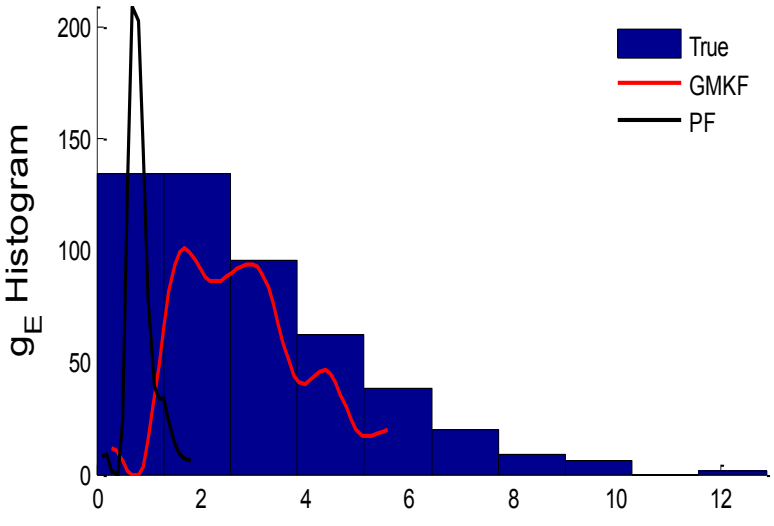


(a)



(b)

Figure 4.6. Estimating synaptic conductances and inputs given a single voltage trace of Example 4.4 using the KF-based (a) and PF-based (b) algorithms. Other descriptions about the figure are the same as those in the Figure 4.4. The initial values of the KF-based algorithm were as follows: the time-varying means (for both excitatory and inhibitory) were generated from a uniform distribution and their variances (for both excitatory and inhibitory) were 5 (for all times). This initial setting (increasing the variance) helped the KF-based algorithm to better estimate the distributions of the synaptic inputs in this experiment.



(a)

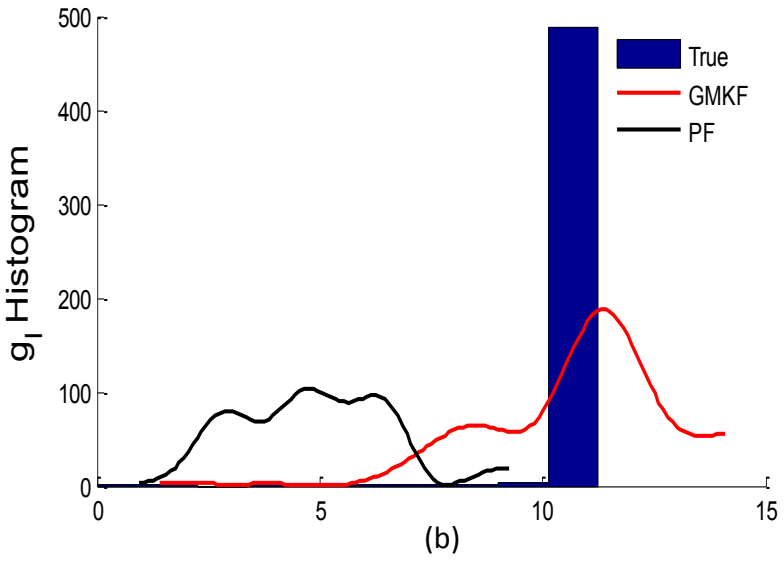
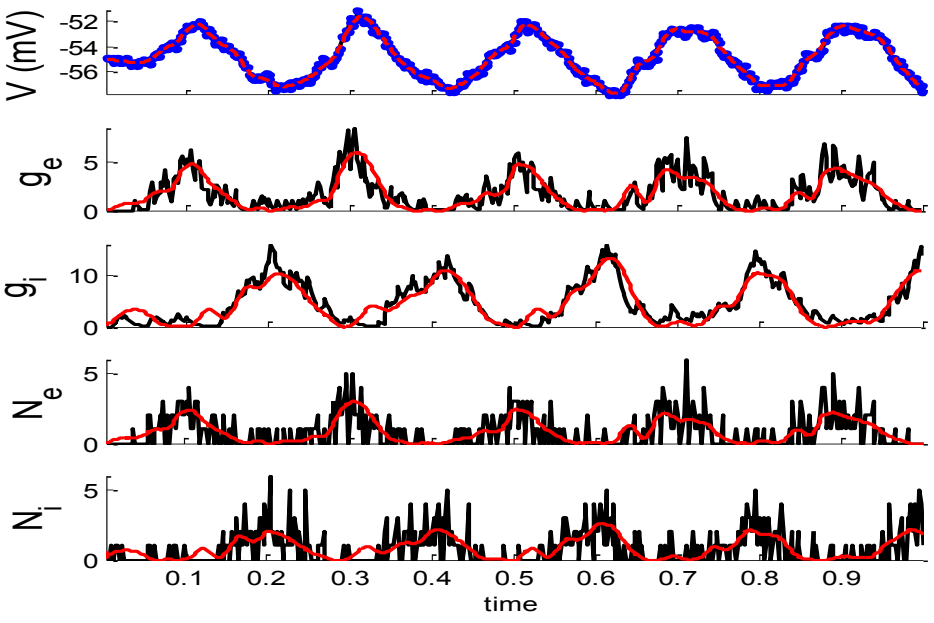


Figure 4.7. Histogram of the excitatory (a) and inhibitory (b) synaptic conductances of the true (blue), estimated using the KF-based (red) and the PF-based (black) algorithms in Example 4.4.

In Example 4.4, we considered an extreme case in which the inhibitory synaptic input had very narrow distribution. In this case, the KF-based algorithm (by selecting appropriate initiation, i.e., large enough variance) could effectively estimate both the excitatory and inhibitory synaptic conductances, even though the PF-based algorithm completely failed (see Figures 4.6 and 4.7). Under this small SNR condition, the prior distribution of synaptic input made an important contribution to the results. While the exponential prior distributions assumed for the PF-based algorithm tended to underestimate the inhibitory synaptic input, the KF-based algorithm could better approximate the inhibitory input by fitting a single Gaussian distribution.

Example 4.5. In this experiment, the specifications of the synaptic inputs were the same as those in Example 4.2. However, the variances of the membrane voltage (σ^2_w) and observation noise (σ^2_ε) increased to 10^{-2} mV and 3 mV, respectively. Figure 4.8 shows the results of the GMKF- and PF-based algorithms in estimating the synaptic conductances from the observed noisy membrane potential generated in this experiment.

The results of each algorithm in Figure 4.8 confirm that the g_E and g_I (and therefore N_E and N_I) are better estimated using the GMKF-based algorithm than by using the PF-based algorithm. It should be noted that the membrane potential is also better tracked using the GMKF-based algorithm. To see how these algorithms approximate the distributions of the excitatory and inhibitory synaptic conductances, we plot the histograms of g_E & g_I estimated by the GMKF- and PF-based algorithms in Figure 4.9.



(a)

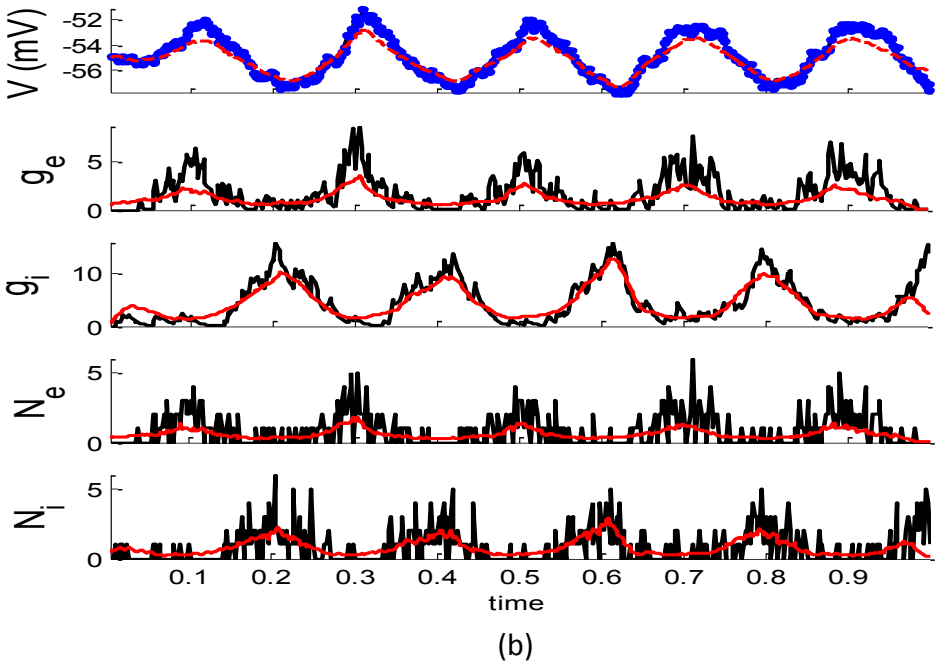
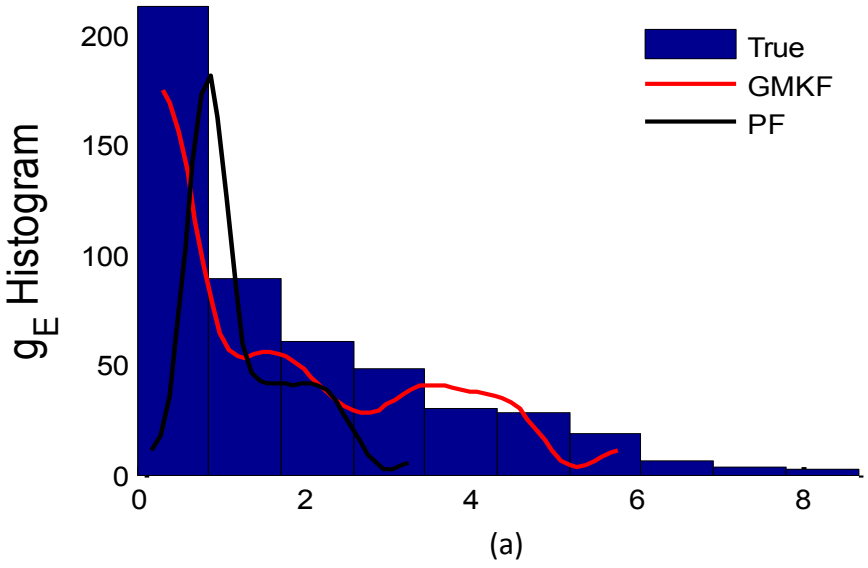


Figure 4.8. Estimating synaptic conductances and inputs given a single voltage trace of Example 4.5 using the GMKF-based (a) and PF-based (b) algorithms. Other descriptions about the figure are the same as those in Figure 4.4. The initial values of the GMKF-based algorithm (*G=2*, *K=4*) were as follows: the time-varying means (for both excitatory and inhibitory) were generated from a uniform distribution and their variances (for both excitatory and inhibitory) were 0.5 for the first mixand and 2 for the second mixand (for all times).



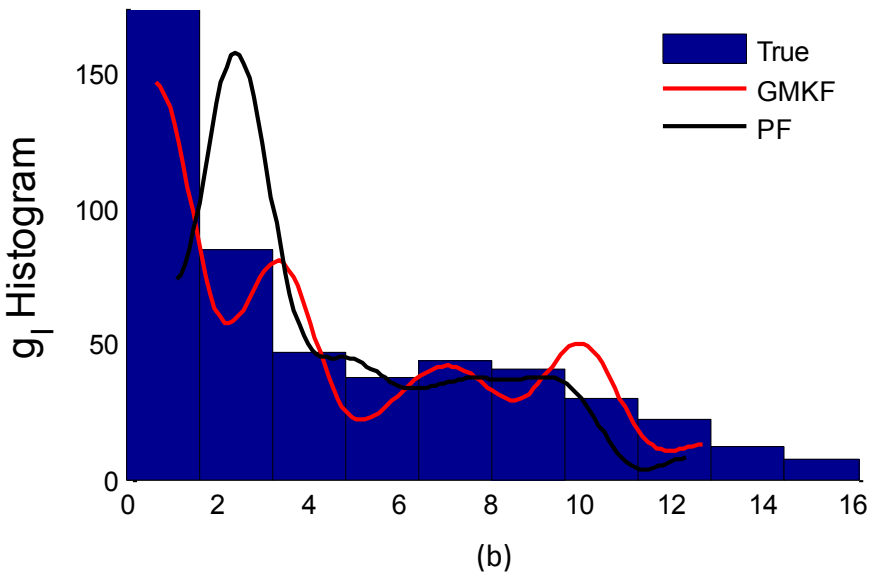
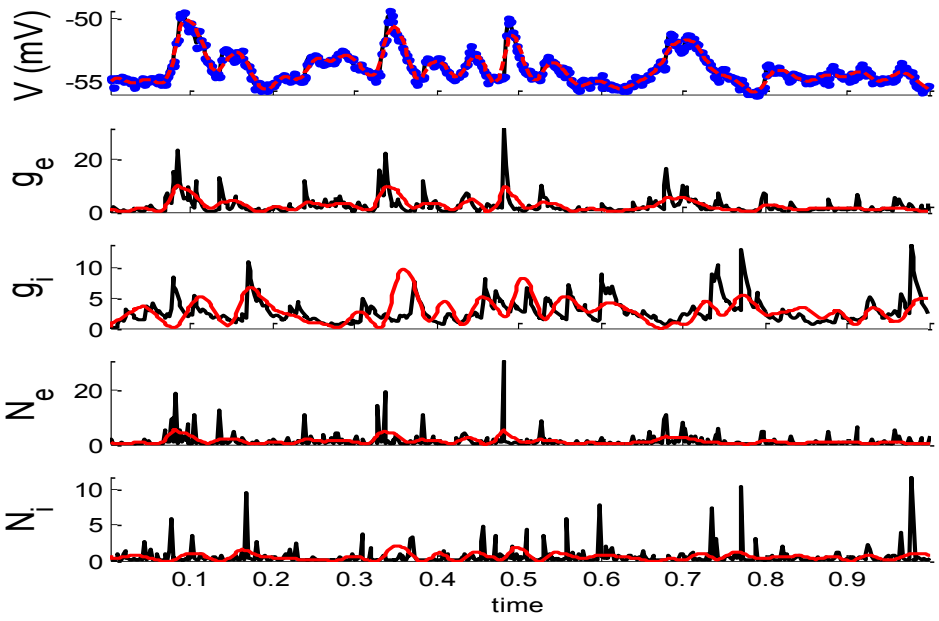


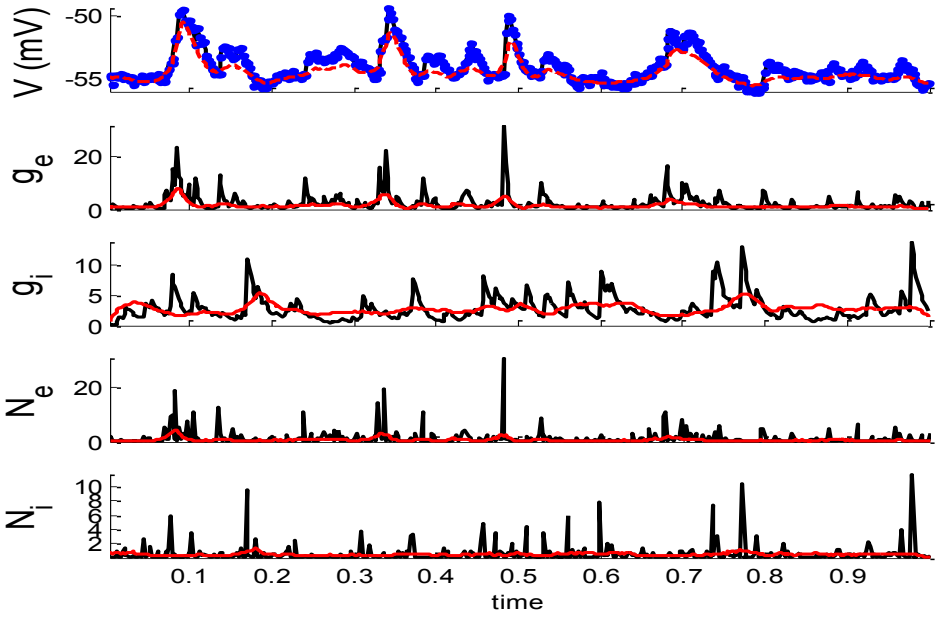
Figure 4.9. Histogram of the excitatory (a) and inhibitory (b) synaptic conductance of the true (blue), estimated using the GMKF-based (red) and PF-based (black) algorithms in Example 4.5.

As can be seen from Figure 4.9, the approximated histogram of the GMKF-based algorithm better represents the true distributions for both the excitatory and inhibitory synaptic conductances.

Example 4.6. In this experiment, the pre-synaptic mean functions are modeled by the absolute value of random realizations of Ornstein-Uhlenbeck processes (same as Example 4.3). The synaptic inputs, both excitatory and inhibitory, are generated from the log-normal distribution of variance 1.5. The variances of the membrane voltage (σ^2_w) and observation noise (σ^2_ε) were 10^{-2} mV and 3 mV, respectively. Figure 4.10 shows the results of the GMKF-based and PF-based algorithms.



(a)



(b)

Figure 4.10. Estimating synaptic conductances and inputs given a single voltage trace of Example 4.6 using the GMKF-based (a) and PF-based (b) algorithms. Other descriptions about the figure were the same as the Figure 4.4. The initial values of the GMKF-based algorithm ($G=2$, $K=4$) were as follows: the time-varying means (for both excitatory and inhibitory) were generated from a uniform distribution and their variances (for both excitatory and inhibitory) were 1 for the first mixand and 4 for the second mixand (for all times).

Similar to Example 4.5 where the GMKF-based algorithm outperformed the PF-based algorithm, Figure 4.10 indicates that the g_E and g_I as well as the membrane voltage are better estimated by the GMKF-based algorithm. The estimated g_E and g_I using the PF-based algorithm could not follow the rapid fluctuations of the synaptic conductances. Figure 4.11 depicts the histogram of the true and estimated g_E & g_I using the GMKF- and PF-based algorithms.

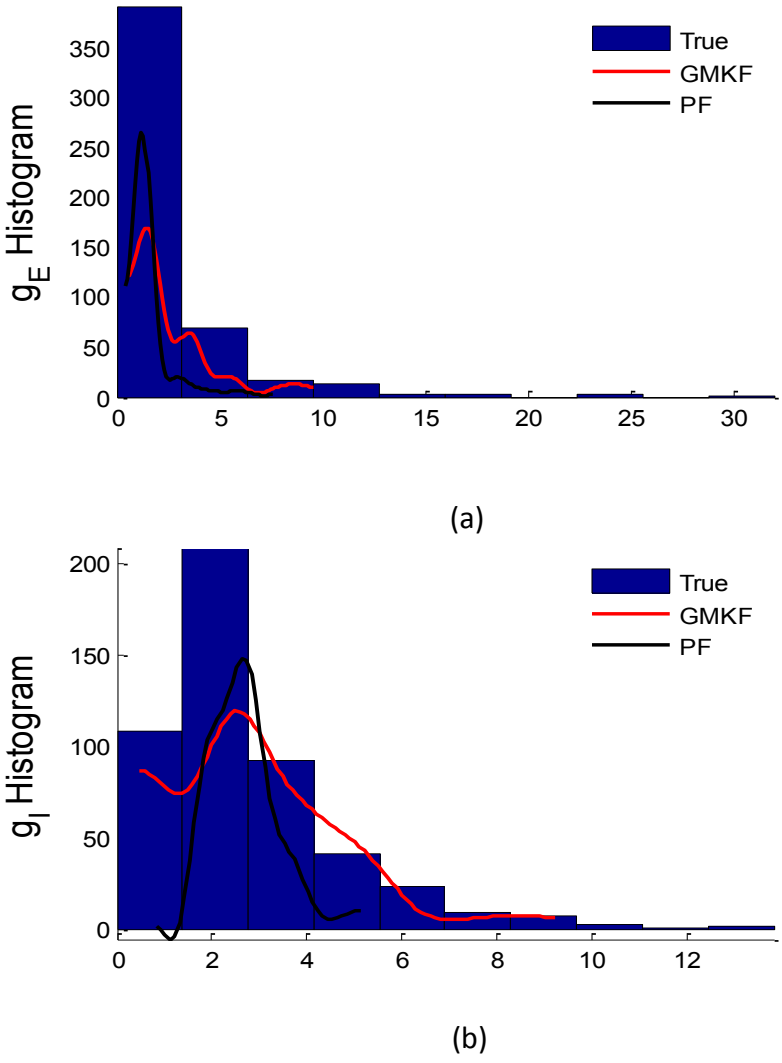


Figure 4.11. Histogram of the excitatory (a) and inhibitory (b) synaptic conductances of the true (blue), estimated by GMKF-basd (red) and PF-based (black) algorithms in Example 4.6.

A heavy high-amplitude tail of the distribution of synaptic inputs has often been observed in neuronal circuits [99-101]. The heavy tail of the log-normal distribution in Example 4.6 (for both g_E and g_I) occasionally produce large synaptic inputs and induced rapid changes in synaptic conductances, which the PF-based algorithm could not keep track of. Hence, this result likely applies to the performance of the GMKF-based vs. PF-based algorithms for heavy-tailed distributions in general. As is clear from Figures 4.10 and 4.11, the GMKF-based algorithm can better track synaptic inputs because GMKF (in this experiment) used two Gaussian mixands that provide more degrees of freedom for fitting the log-normal distribution than only one exponential distribution, which was used in the PF-based algorithm [93].

Theoretically speaking, the PF-based algorithm [93] does not perform accurately under small SNR conditions, if the true underlying distributions for synaptic inputs are different from the presumed prior distributions (e.g., an exponential distribution [93]). Our experiments with various distributions of synaptic inputs confirm that the PF-based algorithm [93] works well if the variance of the observation noise and membrane voltage noise are sufficiently small. The PF-based algorithm can give approximately the same results as the GMKF-based algorithm in this case. However, our experiments suggest that the PF-based algorithm does not accurately estimate synaptic inputs for heavy-tailed distributions (Example 4.6), as well as for distributions that are not properly approximated by the prior distribution (Examples 4.4 and 4.5) in noisy systems. Under this condition, the GMKF-based algorithm outperforms the PF-based algorithm due to its capability of estimating an arbitrary distribution of synaptic inputs by using a GMM. It should be noted that a larger number of mixands (G>2) may be necessary if the synaptic

input distribution is dissimilar to a Gaussian distribution: for example, with a very long tail.

4.5.3 Statistical Analysis

In addition to the above-mentioned observations from the simulation results and in order to compare our algorithms with the PF-based algorithm [93], a statistical analysis is performed in this section. Two types of synaptic inputs, namely, structural (cosine function) and non-structural (O-U process) are considered to generate the membrane potential. Then, each algorithm is applied to 10 trials of these experiments. For the experiment with the structural synaptic input, the same specifications as in Example 4.5 are used and for the experiment in which synaptic inputs are generated from the O-U process, the same specifications as in Example 4.6 are applied. Tables 4.3 and 4.4 quantify the performance of each algorithm in these experiments. For each algorithm, the mean and standard deviation (std) of the normalized error over time are calculated for V, g_E and g_I, where the normalized error is defined as:

$$err(n) = \frac{\sqrt{\sum_{t=1}^{T} \left[x_n(t) - \hat{x}_n(t)\right]^2}}{\sqrt{\sum_{t=1}^{T} x_n(t)^2}}$$

where, x_n and $\hat{x}_n$ are the true and estimated values of the n^{th} trial, respectively. The mean and std are calculated over 10 trials, $err(n)|_{n=1:10}$.

Table4.3: Statistical analysis of the performances of the GMKF-, KF-, and PF-based [93] algorithms in the experiment with structural synaptic input (specifications of the simulation were the same as in Exp 4.5). The values describe trial means and standard deviations of the normalized estimation error.

Algorithms\Features	v	g_E	g_I
PF	$0.0124 \pm 1\times10^{-3}$	$0.5658 \pm 5\times10^{-3}$	$0.3046 \pm 3\times10^{-2}$
KF	$0.0031 \pm 0.2\times10^{-3}$	$0.4106 \pm 0.7\times10^{-3}$	$0.2614 \pm 0.5\times10^{-2}$
GMKF	$0.0033 \pm 0.2\times10^{-3}$	$0.4611 \pm 0.7\times10^{-3}$	$0.2876 \pm 0.5\times10^{-2}$

Table 4.4: Statistical analysis of the performances of the GMKF-, KF- and PF-based [93] algorithms in the experiment with non-structural synaptic input (Specifications of the simulation were the same as in Exp 4.6). Definition of parameters is the same as Table 4.3.

Algorithms\Features	v	g_E	g_I
PF	$0.0246 \pm 7\times10^{-3}$	$0.6678 \pm 0.4\times10^{-2}$	$0.6479 \pm 0.4\times10^{-2}$
KF	$0.0233 \pm 7\times10^{-3}$	$0.6392 \pm 0.6\times10^{-2}$	$0.6322 \pm 0.7\times10^{-2}$
GMKF	$0.0147 \pm 1\times10^{-3}$	$0.4599 \pm 0.6\times10^{-2}$	$0.5811 \pm 0.6\times10^{-2}$

From Tables 4.3 and 4.4, we can conclude that the performance of our KF- and GMKF-based algorithms is better (for all parameters) than that of the PF-based algorithm. When the synaptic distribution is not heavy-tailed (Table 4.3), the KF- and GMKF-based algorithms exhibit approximately the same performance. However, for a heavy-tailed synaptic distribution (log-normal in Table 4.4), the GMKF-based algorithm outperforms the KF-based algorithm. In the GMKF-based algorithm, one could use $G>2$ (number of mixands) which results in more expensive computations. In our simulations, however, $G=2$ was good enough to provide a balance between computational costs and accuracy. For very heavy-tailed distributions, the higher the value of G the better the accuracy obtained for estimating synaptic inputs. Note that the simulations of ($G=2$ and $K=4$, i.e., eight filters for each time) takes approximately the same running time as the PF-based algorithm. Moreover, we observe that K=2, 3 or 4 (number of filters used for estimating $p(\boldsymbol{x}(t)/y(0:t))$) does not change the final results appreciately. As a rule of thumb, we can conclude that $K=G$ is a good choice for selecting the value of K. It should be noted that

when $G=1\&K>1$ (the system noise is approximated by only one Gaussian distribution), it is called Gaussian sum filtering (Kalman or particle can be applied, see [96]). In this case, the conditional probability $p(\boldsymbol{x}(t)|y(0{:}t))$ is estimated using K Gaussian filters. However, our case with $G>1$ and $K>1$ is called Gaussian mixture filtering. In fact, in this case, $G>1$ forces K to be greater than unity in order to better approximate $p(\boldsymbol{x}(t)|y(0{:}t))$, whose filter number grows exponentially over time (see [96]).

Finally, it is to be noted that two other methods have been proposed in the literature for estimating excitatory and inhibitory synaptic conductances from single trials of the recorded membrane potential. Quick alternation of the membrane potential between excitatory and inhibitory reversal potentials [102] enabled nearly simultaneous reconstruction of excitatory and inhibitory synaptic conductances from single trials. One advantage of our method compared to this approach is that it does not require rapid alternations of the membrane potential, which might cause experimental artifacts. Thus, our method provides a wider applicability to existing as well as future experimental data. Another approach is to infer excitatory and inhibitory synaptic conductances by using the oversampling method [103]. Unlike this approach, our KF/GMKF algorithms do not require the manual adjustment of oversampling time steps to suppress singularity problems. The main advantage of our methods in comparison with [38] and [93] relies on the fact that it has the flexibility to estimate an arbitrary (and unknown) probability distribution function of the synaptic inputs by using a GMM.

4.6 Summary

We have proposed in this chapter a recursive algorithm based on GMKF for estimating the excitatory and inhibitory synaptic conductances (and inputs) from noisy recorded membrane potential only. The main advantage of this method in comparison with other recent algorithms relies on the fact that it has the flexibility to estimate an arbitrary (and unknown) probability distribution function of the synaptic inputs by using a GMM. Moreover, we have derived and tested a special case of the GMKF-based algorithm when there is only one mixand, i.e., the Kalman filter, has been derived and tested for estimating the excitatory and inhibitory synaptic conductances. Simulation results have demonstrated the accuracy and robustness of the proposed algorithms in noisy conditions for estimating the synaptic inputs with arbitrary distributions generated from different distributions. In this regard, we have found that the GMKF- and KF-based algorithms outperform the PF-based algorithm. We have also found that the GMKF- and KF-based algorithms have approximately identical performances in many cases, where simple distributions of synaptic inputs are assumed. On the other hand, the GMKF-based algorithms provide much more accurate estimation than the KF-based one when synaptic inputs are drawn from heavy-tailed distributions with many strong synapses. In practice, running both KF-based and GMKF-based algorithms and comparing their results should provide an idea on how complex the underlying distributions of synaptic inputs are. Therefore, the simplicity and high speed of the KF-based algorithm as well as the robustness and general applicability of the GMKF-based algorithm make them efficient techniques for neuroscientists to monitor trial-to-trial variability of the excitatory and inhibitory synaptic inputs.`

Chapter 5

Blind Deconvolution of Hodgkin-Huxley Neuronal Model

5.1 Introduction

A neuron transforms information via a complex interaction between its previous states, its intrinsic properties, and the synaptic input it receives from other neurons. Inferring synaptic input of a neuron only from its membrane potential (output) that contains both sub-threshold and action potentials can effectively elucidate the information processing mechanism of a neuron [6], [36]. In this chapter, the concept of blind deconvolution used in communication and signal processing [104-107] is applied for the first time to the Hodgkin-Huxley (HH) neuronal model [108], to address the problem of reconstructing the hidden dynamics and synaptic input of a single neuron as well as estimating its intrinsic parameters only from a single trace of the noisy membrane potential.

Figure 5.1 shows the schematic representation of encoding the natural stimuli, e.g., an image, in the brain. The challenging question in neuroscience is to understand how this

encoding takes place. Understanding the encoding process without having access to the synaptic inputs of the neurons is impossible. Therefore, inferring the synaptic inputs of a neuron from its output, e.g., the membrane potential in our case, is of great interest in neuroscience.

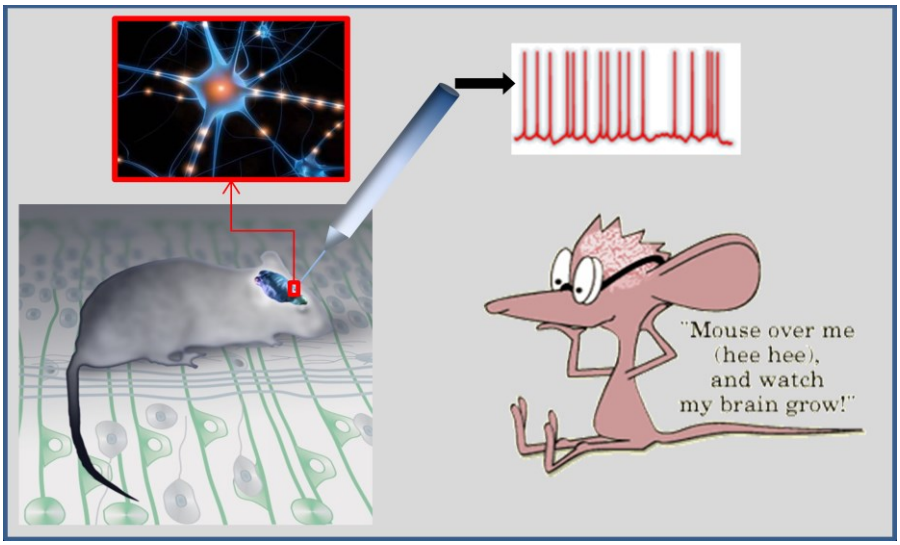


Figure 5.1. Estimating synaptic input and parameters of a neuron from recorded membrane potential

There are several works in neuroscience which aimed to extract the synaptic input of a neuron (excitatory and inhibitory synaptic inputs separately or the sum of them) from the sub-threshold membrane potential (see [87] and references therein). In addition to the significance of synaptic input in neural coding, the dynamics of ion channels influence on neural coding properties [6]. The recent methods in [36] and [37] are the only works in the literature that are not restricted to sub-threshold recordings of the membrane potential wherein the Hodgkin-Huxley (HH) neuronal model is used to represent the behavior of a single neuron and Kalman filtering technique is employed to estimate both the ion

channels dynamics and synaptic input. It is to be noted that the intrinsic parameters of the neuron, e.g., the maximum conductances, are known in [36] and [37].

In this chapter, we extend the scope of the previous works ([36] and [37]) by addressing the problem of reconstructing the hidden dynamics of ion channels and synaptic input (sum of the excitatory and inhibitory) of a single neuron modeled by the HH neuronal model as well as estimating its intrinsic parameters, maximal conductances and statistical parameters (standard deviation of channel noise) from a single trace of noisy membrane potential only. In fact, this chapter can be considered as a generalization of Chapter 3 where not only the parameters of the HH neuronal model are estimated but also the synaptic input of the neuron is inferred from noisy membrane potential only.

5.2 Blind Deconvolution of the Hodgkin-Huxley Neuronal Model

In this section, we briefly describe the problem of blind deconvolution of the HH neuronal model and the assumptions made to address this problem. The HH model has already been introduced in Section 2.3. A modified version of this model including the synaptic input can be stated as follows.

$$C_M \frac{dV}{dt} = -g_{Na} m^3 h (V - E_{Na}) - g_K n^4 (V - E_K) - g_L (V - E_L) + I_{inj}(t) + I_{syn}(t) \qquad (5.1)$$

where (g_{Na}, g_K, g_L) and (E_{Na}, E_K, E_L) denote the maximum conductances and the reversal potentials of the sodium, potassium and leak currents, respectively. I_{syn} is the total

synaptic input (excitatory and inhibitory) that a neuron receives and I_{inj} is the intracellularly injected current. As mentioned in Section 3.8, m, n and h, which indicate the dynamics of the HH model, can be determined by the Langevin equation [82].

$$\frac{dq}{dt} = \alpha_q(V)(1-q) - \beta_q(V)q \ , \ q = m, n, h \tag{5.2}$$

where $\alpha_q(v)$ and $\beta_q(v)$ are nonlinear functions of the voltage (see Chapter 2 for details). In view of the limitations of the imaging techniques, it is impossible to measure all necessary biophysical variables describing a single neuron model. We assume here the intercellular electrophysiological recordings by which the membrane potential, V (plus noise) in (5.1), is the only measurable variable. Our objective is to reconstruct the full HH ionic dynamics, $\{n(t), m(t), h(t)\}$ for the entire recording time, estimating the unknown parameters, $\{g_{Na}, g_K, g_L\}$, and inferring the synaptic input, $I_{syn}(t)$, using solely single trial of membrane potential. This measurement, on the other hand, may contain noise from the recording equipment, which is known as observation noise and modeled by white Gaussian noise [18]. It is to be noted that it is not possible to address this problem using only a single trace of observation (membrane potential) because the number of unknowns overwhelms the number of data points. To overcome this problem some *a priori* knowledge about the unknown variables has to be taken into account. The assumptions are as follows: (1) - the reversal potentials (E_{Na}, E_K, E_L) have been already measured experimentally, (2) - functional form of the voltage-dependent ionic inputs, $\alpha_q(V)$ and $\beta_q(V)$, are known, (3) - similar to [36], the smoothness of the synaptic input is preserved by a random-walk-type prior and (4) - consistent with [30], the initial values of the maximum conductances are randomly selected from the ±25% neighborhood of the true

values to ensure the identifiability of the HH model. Now, to meet our objective based on the aforementioned assumptions, we define a state vector $x = [V, n, m, h, I_{syn}, g_{Na}, g_K, g_L]^{\mathrm{H}}$ including the observed state variable, V, augmented by unobserved state variables, [n, m, h, I_{syn}], and system parameters, [g_{Na}, g_K, g_L]. Therefore, a state space representation of the HH neuron model can be expressed as follows

$$\begin{cases} \dot{\boldsymbol{x}}(t) = F\left[\boldsymbol{x}(t)\right] + BI_{inj}(t) + \boldsymbol{v}(t) \\ y(t) = C\boldsymbol{x}(t) + \varepsilon(t) \end{cases} \tag{5.3}$$

where $C=[1, \boldsymbol{0}_{1\times 7}]$, $B=C^H$, $\varepsilon(t)$ (observation noise), $\boldsymbol{v}(t)$ (system noise) and $I_{inj}(t)$ are mutually independent. ε and $\boldsymbol{v}$ are modeled respectively by a zero-mean white Gaussian noise of variance σ^2_{ε} and that of covariance matrix $\Sigma_{\boldsymbol{v}} = diag([\sigma^2_V, \sigma^2_n, \sigma^2_m, \sigma^2_h, \sigma^2_{syn}, \sigma^2_{Na}, \sigma^2_K, \sigma^2_L])$. $F[\boldsymbol{x}(t)]$ is the time-varying transition function that can be easily obtained from (5.1) and (5.2) (see [18, 37] and Chapter 3 for more details). Let us define $\boldsymbol{\theta} = [\sigma^2_V, \sigma^2_n, \sigma^2_m, \sigma^2_h, \sigma^2_{syn}, \sigma^2_{Na}, \sigma^2_K, \sigma^2_L]^H$ as the statistical parameters of the HH neuronal model. In the next section, we present our proposed recursive algorithm to track (estimate) the state vector $\boldsymbol{x}$ (neuron's dynamics) and the statistical parameter $\boldsymbol{\theta}$.

5.3 Proposed Algorithm

The same framework as described in Section 4.3 (see also Figure 4.2) is used here to develop a recursive algorithm for estimating the dynamics and parameter of (5.1). According to this framework, the recursive algorithm consists of three main steps whose implementations for the blind deconvolution problem are given here. It begins with the initial values, $\boldsymbol{\theta}_0$, which can be set to very small values like 10^{-6} (for all statistical parameters).

Step 1. In this step, the extended Kalman filtering (EKF) is employed to accomplish filtering the states $\boldsymbol{x}$ and providing the first and second order conditional statistics, $\mathrm{E}\{\boldsymbol{x}(t)\,|y_{0:t}\}$ and $\mathrm{E}\{\boldsymbol{x}(t)\,(\boldsymbol{x}(t))^{H}|y_{0:t}\}$ (full derivation of EKF for the HH model is given in Chapter 3).

Step 2. Sufficient statistics for the EM algorithm, $\mathrm{E}\{\boldsymbol{x}(t)|y_{0:T}\}$ and $\mathrm{E}\{\boldsymbol{x}(t)\,(\boldsymbol{x}(t))^{H}\,|y_{0:T}\}$ over the entire time, $\{0{:}T\}$, are calculated by Kalman backward filtering (smoothing), in this step.

Step 3. Here, the new estimation of the statistical parameters, $\hat{\boldsymbol{\theta}}$, is calculated by the EM algorithm as follows.

$$\begin{aligned}\max_{s.t.\hat{\theta}} Q(\theta,\hat{\theta}) &= \max\left(E\left\{\log\left(p(Y,X\,|\,\hat{\theta})\right)|\,Y,\theta\right\}\right)\\ &= \max\left(\int \log\left(p(Y,X\,|\,\hat{\theta})\right)p(X\,|\,Y)dX\right)\end{aligned} \tag{5.4}$$

where Y and X stand for the observation, $\{y\}_{0:T}$, and states, $\{\boldsymbol{x}\}_{0:T}$, over the entire time. In fact, the aim is to find new statistical parameters $\hat{\boldsymbol{\theta}}$ that maximizes the expected joint log

likelihood of the observation and the hidden states with respect to statistical parameters $\boldsymbol{\theta}$. Expanding (5.4), we can write:

$$E\left\{\log\left(p(Y,X\mid\hat{\theta})\right)\mid Y,\theta\right\}=E_{p(X|Y,\theta)}\left\{\sum_{t=1}^{T}\log(\,p(y(t)\mid \boldsymbol{x}(t),\theta))\right.$$
$$\left.+\sum_{t=2}^{T}\log(\,p(\boldsymbol{x}(t)\mid \boldsymbol{x}(t-1),\theta))+\log(\,p(\boldsymbol{x}(1)\mid\theta))\right\} \tag{5.5}$$

In this chapter, similar to Chapter 4, we treat time t as a discrete variable that takes integer values between *1* and T, where $T\times dt$ is the entire (physical) time of recording. Recalling that $p(y(t)|\boldsymbol{x}(t),\boldsymbol{\theta}) = N(y(t);C\boldsymbol{x}(t)\,,\sigma^2_{\varepsilon})$ and $p(\boldsymbol{x}(t)|\,\boldsymbol{x}(t\text{-}1),\boldsymbol{\theta}) = N(\boldsymbol{x}(t);F[\boldsymbol{x}(t)]+BI_{inj}\,,\Sigma_{\boldsymbol{v}})$, where $N(\mu,\sigma^2)$ stands for the normal distribution of mean μ and variance σ^2, and taking the derivative of (5.5) with respect to $\Sigma^{-1}_{\boldsymbol{v}}$ (=$diag(\boldsymbol{\theta})$), we can calculate the new estimate of $\Sigma_{\boldsymbol{v}}$ as follows.

$$\frac{\partial Q(\theta,\hat{\theta})}{\partial\hat{\Sigma}_{\boldsymbol{v}}^{-1}}=\frac{1}{2}\sum_{t=2}^{T}\left(\hat{\Sigma}_{\boldsymbol{v}}-\left(\hat{\boldsymbol{x}}^{t}(t)-A(t)\hat{\boldsymbol{x}}^{t-1}(t)\right)\left(\hat{\boldsymbol{x}}^{t}(t)-A(t)\hat{\boldsymbol{x}}^{t-1}(t)\right)^{H}\right)=0$$
$$\Rightarrow\hat{\Sigma}_{\boldsymbol{v}}=\frac{\sum_{t=2}^{T}\left(\hat{R}_{x}^{t}(t)-A(t)\hat{R}_{x}^{t-1}(t)^{H}\right)}{T-1} \tag{5.6}$$

where $\hat{R}^{t}_{\boldsymbol{x}}(t)=\hat{\Sigma}^{t}_{x}(t)+\hat{\boldsymbol{x}}^{\boldsymbol{t}}(t)\left(\hat{\boldsymbol{x}}^{\boldsymbol{t}}(t)\right)^{H}$, $\hat{R}^{t-1}_{x}(t)=\hat{\Sigma}^{t-1}_{\boldsymbol{x}}(t)+\hat{\boldsymbol{x}}^{\boldsymbol{t}}(t)\left(\hat{\boldsymbol{x}}^{\boldsymbol{t\text{-}1}}(t-1)\right)^{H}$.

In the above, $\hat{\boldsymbol{x}}^{\boldsymbol{t}}(t)=E\{\boldsymbol{x}(t)\mid y_{0:T}\}$ and $\hat{\Sigma}^{t}_{x}(t)=E\{\boldsymbol{x}(t)\boldsymbol{x}(t)^{H}\mid y_{0:T}\}$ are the first and second order statistics that have been already calculated in the Kalman smoothing step (see Appendix E for more detail). Then, $\boldsymbol{\theta}$ can be easily obtained as the diagonal elements of $\hat{\Sigma}_{\boldsymbol{v}}$. It is to

be noted that $A(t)$ in (5.6) represents the first order derivative of the transition function $\boldsymbol{F}$ with respect to the states, $\boldsymbol{x}(t)$ (see Appendix A for computing $A(t)$).

Once $\boldsymbol{\theta}$ is calculated, it would be considered as the initial value for the recursive algorithm. The algorithm stops when no considerable change occurs in two consecutive iterations. It is observed from our simulations that the variation of the estimated synaptic input, I_{syn}, is a good candidate to be used as the stopping criterion. Therefore, our algorithm stops when the variance of difference of the estimated synaptic inputs in the consecutive iterations is less than 5%.

5.4 Performance of the Proposed Algorithm

In this section, several illustrative numerical simulations are provided to study the accuracy of our proposed algorithm for blind deconvolution of the HH neuronal model. Two different types of synaptic input are considered in our simulations. In the first simulation, the synaptic input contains two jumps (see Figure 5.2) which do not fulfill the smoothness assumption we previously made. The second simulation is more realistic wherein the synaptic input is generated from Ornstein-Uhlenbeck process (white Gaussian noise filtered by $0.4/(1-0.9z^{-1})$). For each experiment, the accuracy of our proposed algorithm in estimating the parameters of the HH neuronal model and reconstructing its synaptic input is demonstrated. The simulated data is generated by an HH model whose specifications are: $\{E_{Na}=55,\ E_K=-90,\ E_L=-70\}^{mV}$, $\{g_{Na}=32,\ g_K=10,\ g_L=0.1\}^{\mu S/m2}$, $c_M=1^{\mu F/cm2}$ and the rate constants of the ion channel state transitions ($\alpha_q(v)$ and $\beta_q(v)$) are the same as those used in Equation (2.7). A zero mean white Gaussian noise of standard deviation 10 mV is added to the generated membrane potential as the

observation noise. All the simulations are carried out by MATLAB and the HH neuron model dynamics are obtained by using the "ode15" of MATLAB functions with 0.01 ms as the integration time step while the membrane potential is sampled every 0.1 *ms*. Figures 5.2 and 5.5 show the noisy recorded membrane potentials (top) and the true synaptic inputs (bottom) for the first and the second simulation, respectively. Figures 5.3 and 5.6, respectively, demonstrate the reconstructed versus the true membrane potentials (top) and the synaptic input (bottom) of each experiment using the proposed algorithm. Moreover, Figures 5.4 and 5.7 illustrate the reconstruction of the HH channel dynamics, for each experiment.

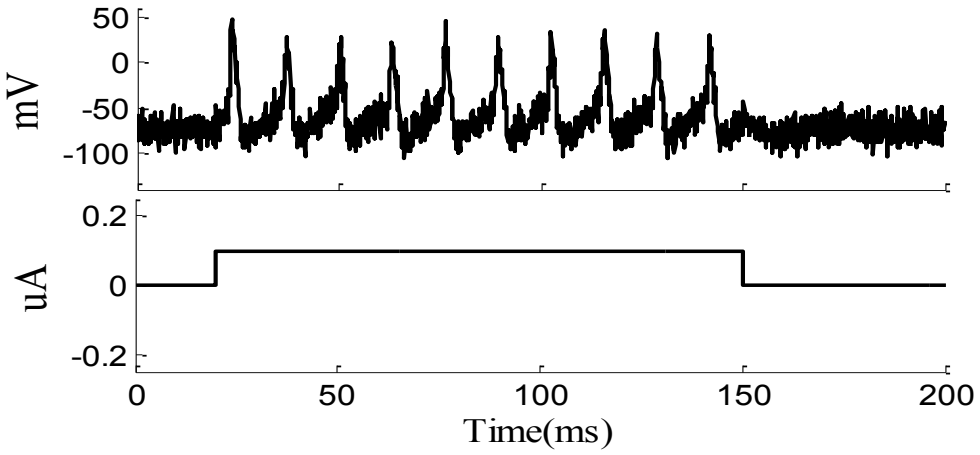


Figure 5.2. Noisy membrane potential in the first simulation (top) and original synaptic input (bottom).

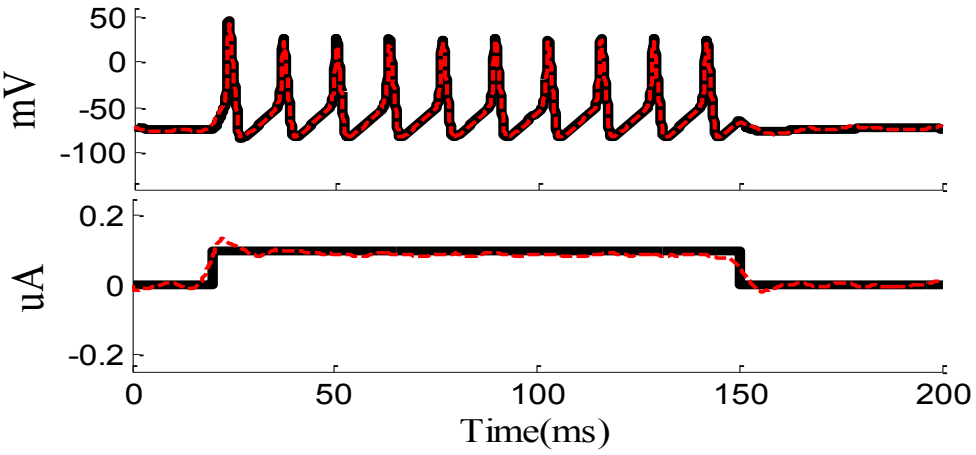


Figure 5.3. Estimated (red dashed line) versus true (black solid) membrane potential (top) and synaptic input (bottom). The algorithm stops in the 7th iteration. The initial value of θ (for all variables) in this example is set to 10^{-6}.

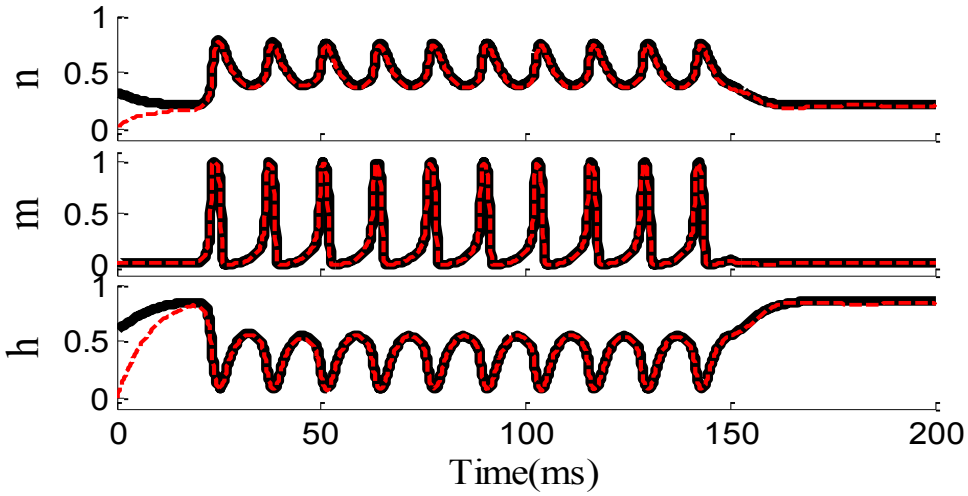


Figure 5.4. Estimated (red dash line) versus true (black solid) channel dynamics of the HH model. The initial values of n, m, h are zero.

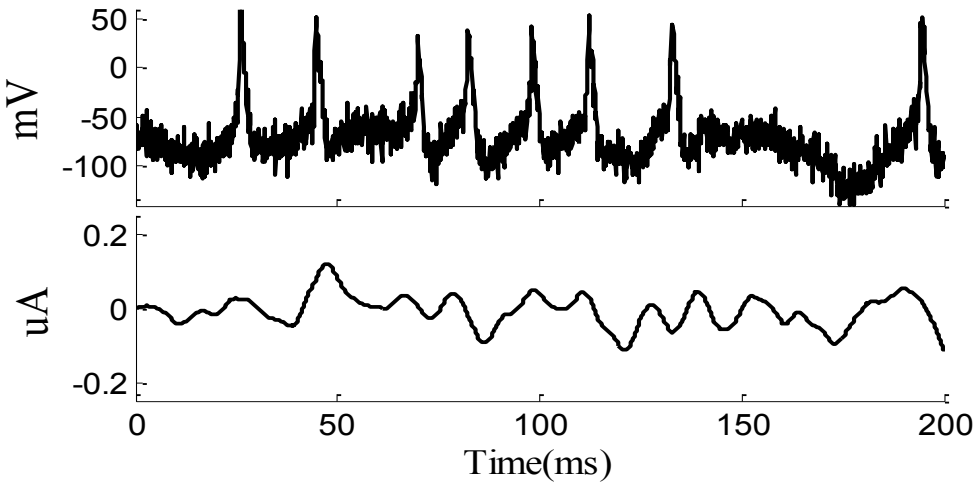


Figure 5.5. Noisy membrane potential in the second simulation (top) and original synaptic input (bottom). The step current, I_{inj} = 0.06 μA/cm^2, $20ms \leq t \leq 150ms$, is injected to neuron in this simulation.

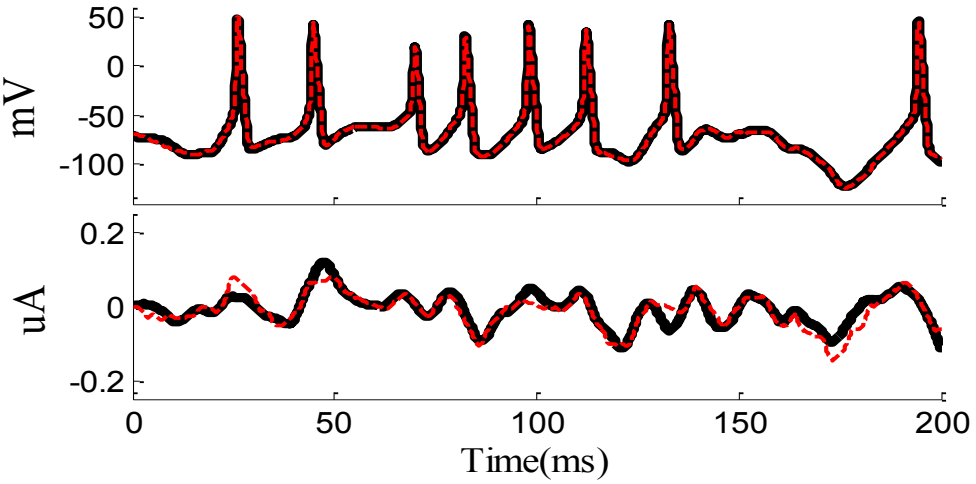


Figure 5.6. Estimated (red dashed line) versus true (black solid) membrane potential (top) and synaptic input (bottom). The algorithm stops in the 3th iteration. Synaptic input is generate by low pass filtering ($0.4/(1-0.9z^{-1})$) the white Gaussian noise. The initial value of θ (for all variables) in this example is set to 10^{-4}.

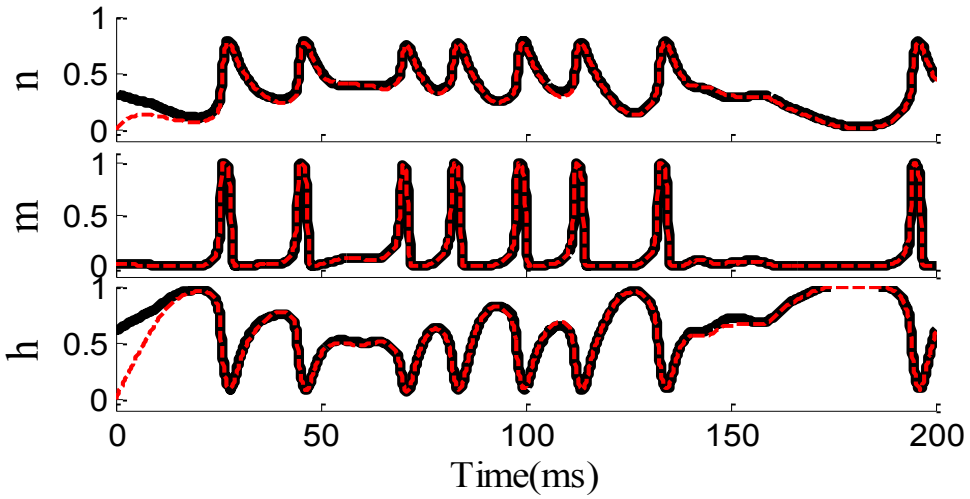


Figure 5.7. Estimated (red dash line) versus true (black solid) channel dynamics of the HH model. The initial values of n, m, h are zero.

Figure 5.8 shows as to how the parameters of the HH model, the maximum conductances in the first experiment, converge to their true values. As can be seen from this figure, all the estimated parameters, g_{Na}, g_K and g_L accurately converge to their true values. However, as mentioned in Chapter 3, the estimated g_{Na} is biased. This might be due to the observation noise that the algorithm underestimate the sharpness of spikes and therefore underestimates the sodium current.

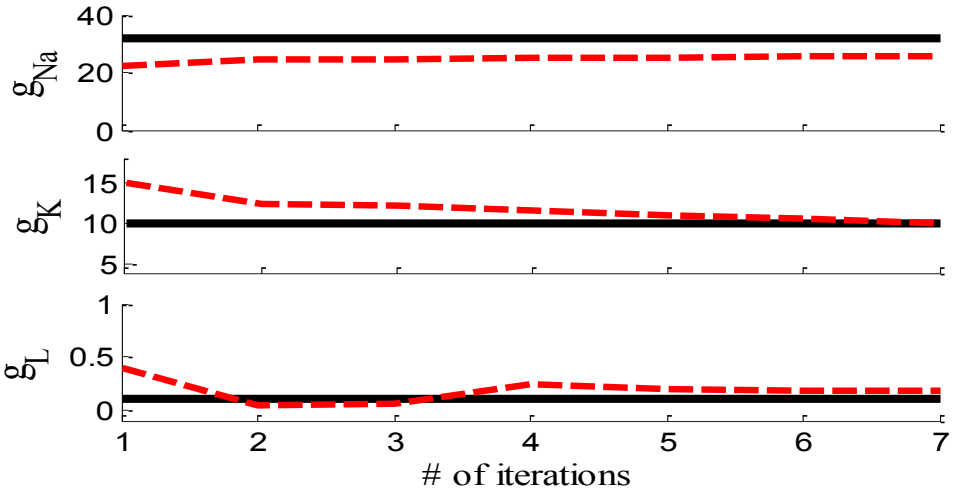


Figure 5.8. Estimated (red dash line) versus true (black solid) parameters of the HH model.

As seen in Figures 5.2 – 5.8, the membrane potential, V, the dynamics of ion channels, $\{n, m, h\}$, the synaptic input, I_{syn}, and the intrinsic parameters $\{g_{Na}, g_K, g_L\}$, are all estimated with excellent accuracy only from noisy recorded membrane voltage using our proposed algorithm.

The importance of our technique in blind deconvolution of the HH neuronal model relies on the fact that the unobserved synaptic input of a neuron can be estimated from a single trial of recorded membrane potential. It is to be highlighted that the capability of trail-to-trial inferring the synaptic inputs of a single neuron is of high interest in neuroscience because the stochastic behavior of neurons can be assessed. To the best of our knowledge, our proposed technique is the first algorithm for trial-to-trial inferring synaptic input of a spiking neuron. In fact, the proposed technique is a generalization of the algorithms employed in Chapters 3 and 4. The concept of the JEKF algorithm that has been already developed in Chapter 3 is used in the second step of the proposed algorithm

to jointly estimate the parameters and the states of the HH neuronal model. Moreover, since the synaptic input of the neuron is considered as a new state for our algorithm, inferring this state reminds the inference of the excitatory and inhibitory synaptic inputs in Chapter 4. It is also worth mentioning that the proposed algorithm for blind deconvolution of the HH model is based on the framework used in Chapter 4.

5.5 Summary

A novel recursive algorithm has been proposed, in this chapter, to address the problem of estimating the unobserved dynamics, the synaptic input and the intrinsic parameters of the Hodgkin-Huxley neuronal model. Application of the *blind deconvolution* to the HH neuronal model has been tackled by employing an extended Kalman filtering followed by an EM algorithm. The robustness and accuracy of this algorithm have been validated by two illustrative simulations. The corresponding promising results imply that the proposed algorithm provides a powerful framework for estimating the unobserved dynamics and input of a neuron and therefore, can better reveal how neurons transform information from the synaptic input to the membrane potential.

Chapter 6

Conclusion

6.1 Concluding Remarks

Investigation of some of the open problems in computational neuroscience that helps in our understanding about the information processing of the neurons has been the core objective of this thesis. To meet this objective, three challenging problems in the computational neuroscience that aim to identify the hidden dynamics, estimate the intrinsic parameters, and reconstruct the synaptic input of a single neuron have been studied. Several methods based on Bayesian signal processing have been developed to address these problems.

In the first part of this study, Kalman filtering (KF) algorithms for the most detailed biophysical model representing the behavior of a single neuron, the Hodgkin-Huxley (HH) neuronal model, have been developed to identify the hidden dynamics and estimate the unknown parameters of a single neuron. Since there are two types of unknowns, namely, the time-varying dynamics and the constant parameters, that have to be estimated

in the HH neuronal model, the unscented and extended Kalman filtering algorithms have been used in conjunction with two parameter estimation methods, namely, the dual and joint estimation methods, leading to four KF algorithms, namely, joint unscented KF (JUKF), dual unscented KF (DUKF), joint extended KF (JEKF) and dual extended KF (DEKF). The accuracy of these four algorithms as well as their speeds and computational costs have been studied in order to determine the most appropriate KF-based algorithm in real world applications such as dynamic clamp. The accuracy of these four proposed algorithms has been evaluated under different signal to observation noise ratios. The results have shown that the EKF-based methods perform as well as JUKF does, especially at low SNRs. More importantly, the faster implementation of the EKF methods makes them particularly useful in real time applications. Moreover, JUKF and JEKF have been extended to estimate the entire parameters of the HH neuronal model including the kinetics of the ion channels. In order to verify the feasibility of these methods, two experiments have been conducted. The accuracy of the developed JUKF and JEKF algorithms has been confirmed in our simulation study. Furthermore, by applying these algorithms to real data, the entire kinetics of the ion channels have been inferred. More importantly, the hidden dynamics of the real neuron which represent the dynamics of the ion channels have been reconstructed by using the proposed methods.

In the second part of the thesis, the problem of inferring the excitatory and inhibitory synaptic inputs of a single neuron from a single trial of the recorded membrane potential has been studied. Due to the fact that most of the existing methods in the literature are based on least square (LS) estimation, they can only estimate the trial mean of the synaptic inputs. Moreover, it has been established that the LS estimation technique is a

biased-estimator for the excitatory and inhibitory synaptic inputs. As an alternative for the LS method, a novel recursive algorithm based on Gaussian mixture Kalman filtering (GMKF) has been proposed to estimate both the synaptic inputs from a single trial of the noisy membrane potential. Moreover, the simplest case of our algorithm which has only one Gaussian mixand, namely, the well-known Kalman filtering has been developed in a recursive framework, yielding simpler, faster, and more robust results than the existing particle filtering based algorithms do. In order to compare the performance of the proposed algorithms with that of other existing algorithms, two illustrative simulation-cases have been studied. In the first case, it has been demonstrated that the LS-based methods are not capable of estimating the excitatory and inhibitory synaptic inputs from a single trial of the recorded data, while the proposed algorithms and the PF method [93] can accurately estimate these inputs from a single trace of the membrane potential. In the second case, the performance of our KF- and GMKF- based algorithms has been compared with that of PF-based method. From the simulation studies, it was observed that when SNR is high, estimating the excitatory and inhibitory synaptic inputs was relatively easy and the results do not depend much on the algorithms used. However, in the case of low SNRs, the proposed algorithms have significantly outperformed the PF algorithm. In conclusion, the proposed GMKF-based algorithm has exhibited accurate and robust performance over the entire range of parameters studied and our KF-based algorithm has exhibited fast and simple estimation in many representative scenarios. Thus, our proposed algorithms can be considered to be promising tools in neuroscience for estimating the excitatory and inhibitory synaptic conductances, or equivalently synaptic inputs, from a single trial of the recorded membrane potential.

In the third part, the problem of identifying the synaptic input of a single neuron as well as its intrinsic parameters and hidden dynamics has been studied. This problem has been tackled by applying the concepts of blind deconvolution to the HH neuronal model, and accomplished via a novel recursive algorithm based on the extended KF, followed by the expectation-maximization (EM) algorithm that estimates the statistical parameters of the HH neuronal model. Since the proposed algorithm is recursive, the estimated parameters act as the initial values for the next iteration. The accuracy and robustness of the proposed algorithm have been demonstrated through numerical simulations. The proposed algorithm should be particularly useful in understanding of the neural coding mechanism of a neuron.

6.2 Scope for Future Work

Considering that a number of Bayesian-based approaches have been proposed in this thesis to address some of the open problems in computational neuroscience, several extensions on the proposed algorithms that should make them more useful to deal with real data are worth studying further. Moreover, in view of the signal processing theme of the proposed algorithms and their promising performance for problems with highly nonlinear and time-varying structures, further investigation on the application of the proposed methods in other engineering areas is suggested.

As mentioned in Chapter 3, four KF-based algorithms have been derived for the HH neuronal model with known kinetics. Then, JUKF and JEKF algorithms have been extended to infer the entire kinetics of the HH neuronal model. The kinetics can be

measured in real neurons through very complicated electrophysiological experiments, hence it is important to compare the results of the proposed algorithms with those measured experimentally. Moreover, it is worth improving the propose KF algorithms to the most recent ones such as cubature KF (CKF) [109] and adaptive unscented KF [110].

The proposed GMKF- and KF- based algorithms in Chapter 4 can be generalized for the spiking neurons. Although the main experimental tests in neuroscience for inferring excitatory and inhibitory synaptic inputs have been carried out on the neurons whose active channels are pharmacologically blocked, it is worth improving our proposed recursive framework for the spiking neuronal models, such as the Moris-Lecar and HH models. It is obvious that trial-to-trial tracking of the synaptic inputs, which has been addressed by using our algorithms, can better elucidate the functioning roles of the excitatory and inhibitory synaptic inputs. Thus, it is highly recommended that the proposed methods be used in experimental investigations.

Our proposed algorithm in Chapter 5 that uses the concept of blind deconvolution to estimate both the dynamics and the input of a single neuron can be enhanced in a way similar to that mentioned for the algorithms in Chapter 3. This algorithm can be improved in order to estimate the kinetics of the HH neuronal model. Furthermore, an algorithm can be developed to estimate both the excitatory and inhibitory synaptic inputs.

References

[1] A. P. Alivisatos, M. Chun, G. M. Church, R. J. Greenspan, M. L. Roukes and R. Yuste, "The brain activity map project and the challenge of functional connectomics," *Neuron,* vol. 74, pp. 970-974, Jun 21, 2012.

[2] The BRAIN Initiative (Brain Research through Advancing Innovative Neurotechnologies), "http://en.wikipedia.org/wiki/BRAIN_Initiative," .

[3] H. Markram, "The blue brain project," *Nat. Rev. Neurosci.,* vol. 7, pp. 153-160, Feb, 2006.

[4] The Human Brain Project (HBP), "http://www.humanbrainproject.eu/," .

[5] Computatiopnal Neuroscience, "http://en.wikipedia.org/wiki/Computational_neuroscience," .

[6] C. Koch, Ed., *Biophysics of Computation.* Oxford University Press, 1999.

[7] J. S. Coggan, T. M. Bartol, E. Esquenazi, J. R. Stiles, S. Lamont, M. E. Martone, D. K. Berg, M. H. Ellisman and T. J. Sejnowski, "Evidence for ectopic neurotransmission at a neuronal synapse," *Science,* vol. 309, pp. 446-451, Jul 15, 2005.

[8] C. K. Machens, R. Romo and C. D. Brody, "Flexible control of mutual inhibition: a neural model of two-interval discrimination," *Science,* vol. 307, pp. 1121-1124, Feb 18, 2005.

[9] F. Crick and C. Koch, "A framework for consciousness," *Nat. Neurosci.,* vol. 6, pp. 119-126, Feb, 2003.

[10] B. J. Baker, E. K. Kosmidis, D. Vucinic, C. X. Falk, L. B. Cohen, M. Djurisic and D. Zecevic, "Imaging brain activity with voltage- and calcium-sensitive dyes," *Cell. Mol. Neurobiol.,* vol. 25, pp. 245-282, Mar, 2005.

[11] D. Smetters, A. Majewska and R. Yuste, "Detecting action potentials in neuronal populations with calcium imaging," *Methods,* vol. 18, pp. 215-221, Jun, 1999.

[12] Astrid A. Prinz and Robert H. Cudmore, "Dynamic clamp," *Scholarpedia,* 2011.

[13] J. C. Bettencourt, K. P. Lillis, L. R. Stupin and J. A. White, "Effects of imperfect dynamic clamp: computational and experimental results," *J. Neurosci. Methods,* vol. 169, pp. 282-289, Apr 30, 2008.

[14] A. A. Prinz, L. F. Abbott and E. Marder, "The dynamic clamp comes of age," *Trends Neurosci.,* vol. 27, pp. 218-224, Apr, 2004.

[15] A. A. Sharp, M. B. O'Neil, L. F. Abbott and E. Marder, "The dynamic clamp: artificial conductances in biological neurons," *Trends Neurosci.,* vol. 16, pp. 389-394, Oct, 1993.

[16] K. Deisseroth, G. Feng, A. K. Majewska, G. Miesenbock, A. Ting and M. J. Schnitzer, "Next-generation optical technologies for illuminating genetically targeted brain circuits," *J. Neurosci.,* vol. 26, pp. 10380-10386, Oct 11, 2006.

[17] K. Chung and K. Deisseroth, "CLARITY for mapping the nervous system," *Nat. Methods,* vol. 10, pp. 508-513, Jun, 2013.

[18] G. Ullah and S. J. Schiff, "Tracking and control of neuronal Hodgkin-Huxley dynamics," *Physical Review.E, Statistical, Nonlinear, and Soft Matter Physics,* vol. 79, pp. 040901, 2009.

[19] James V. Candy, Ed., *Model-Based Signal Processing.* IEEE Press, 2006.

[20] James V. Candy, Ed., *Bayesian Signal Processing; Classical, Modern, and Particle Filtering Methods.* John Wiley & Sons Inc, 2009.

[21] D. V. Vavoulis, V. A. Straub, J. A. Aston and J. Feng, "A self-organizing state-space-model approach for parameter estimation in hodgkin-huxley-type models of single neurons," *PLoS Comput. Biol.,* vol. 8, pp. e1002401, 2012.

[22] G. Ullah and S. J. Schiff, "Assimilating seizure dynamics," *PLoS Comput. Biol.,* vol. 6, pp. e1000776, May 6, 2010.

[23] L. Buhry, F. Grassia, A. Giremus, E. Grivel, S. Renaud and S. Saighi, "Automated parameter estimation of the Hodgkin-Huxley model using the differential evolution algorithm: application to neuromimetic analog integrated circuits," *Neural Comput.,* vol. 23, pp. 2599-2625, Oct, 2011.

[24] Buhry. L, Pace. M, Saighi. S, "Global parameter estimation of an Hodgkin–Huxley formalism using membrane voltage recordings: Application to neuro-mimetic analog integrated circuits," *Neurocomputing,* vol. 85, 2012.

[25] A. R. Willms, "NEUROFIT: software for fitting Hodgkin-Huxley models to voltage-clamp data," *J. Neurosci. Methods,* vol. 121, pp. 139-150, Dec 15, 2002.

[26] C. Rossant, D. F. Goodman, J. Platkiewicz and R. Brette, "Automatic fitting of spiking neuron models to electrophysiological recordings," *Front. Neuroinform,* vol. 4, pp. 2, Mar 5, 2010.

[27] N. Keren, N. Peled and A. Korngreen, "Constraining compartmental models using multiple voltage recordings and genetic algorithms," *J. Neurophysiol.,* vol. 94, pp. 3730-3742, Dec, 2005.

[28] M. C. Vanier and J. M. Bower, "A comparative survey of automated parameter-search methods for compartmental neural models," *J. Comput. Neurosci.,* vol. 7, pp. 149-171, Sep-Oct, 1999.

[29] W. Van Geit, P. Achard and E. De Schutter, "Neurofitter: a parameter tuning package for a wide range of electrophysiological neuron models," *Front. Neuroinform,* vol. 1, pp. 1, 2007.

[30] David Csercsik, KatalinM.Hangos, Gabor Szederkenyi, "Identifiability analysis and parameter estimation of a single Hodgkin–Huxley type voltage dependent ion channel under voltage step measurement conditions," *Neurocomputing,* vol. 77, pp. 178-188, 2012.

[31] S. Druckmann, Y. Banitt, A. Gidon, F. Schurmann, H. Markram and I. Segev, "A novel multiple objective optimization framework for constraining conductance-based neuron models by experimental data," *Front. Neurosci.,* vol. 1, pp. 7-18, Nov, 2007.

[32] K. Doya, A. I. Selverston and P. F. ROwat, "A Hodgkin–Huxley type neuron model that learns slow non-spike oscillations," *Proceedings of NIPS,* 1993.

[33] M. Pospischil, M. Toledo-Rodriguez, C. Monier, Z. Piwkowska, T. Bal, Y. Fregnac, H. Markram and A. Destexhe, "Minimal Hodgkin-Huxley type models for different classes of cortical and thalamic neurons," *Biol. Cybern.,* vol. 99, pp. 427-441, Nov, 2008.

[34] G. Riget and J. S. Vesterstrom, "A diversity-guided particle swarm optimizer—the arPSO," Dept. Computer Science, University of Aarhus, Denmark, Tech. Rep. Technical report, EVA life, 2002.

[35] M. Lankarany, W. P. Zhu, M. N. S. Swamy and T. and Toyoizumi, "Trial-to-trial tracking of excitatory and inhibitory synaptic conductances using gaussian mixture kalman filtering," in *22nd Annual Computational Neuroscience Meeting (CNS),* Paris, France, 2013, .

[36] Kobayashi. R, Tsubo. Y, Lansky. P and Shinomoto. S, "Estimating time-varying input signals and ion channel states from a single voltage trace of a neuron," *Advances in Neural Information Processing Systems 24 (NIPS 2011),* 2011.

[37] Yina Wei, Ghanim Ullah, Ruchi Parekh, Jokubas Ziburkus, Steven J. Schiff, "Kalman filter tracking of intracellular neuronal voltage and current," in *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC),* Orlando, FL, USA, 2011, .

[38] R. Kobayashi, S. Shinomoto and P. Lansky, "Estimation of time-dependent input from neuronal membrane potential," *Neural Comput.,* vol. 23, pp. 3070-3093, Dec, 2011.

[39] Q. J. Huys and L. Paninski, "Smoothing of, and parameter estimation from, noisy biophysical recordings," *PLoS Comput. Biol.,* vol. 5, pp. e1000379, May, 2009.

[40] Doucet A, Godsill S, Andrieu C, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Stat Comput,* vol. 10, pp. 197-208, 2000.

[41] Dempster A, Laird N, Rubin D, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J Royal Stat Soc Series B (Methodolical),* 1977.

[42] S. J. Schiff and T. Sauer, "Kalman filter control of a model of spatiotemporal cortical dynamics," *J. Neural Eng.,* vol. 5, pp. 1-8, Mar, 2008.

[43] James B. Maciokas, "Towards Understanding of the synergistic Properties of Cortical Processing: A Neuronal Computation Modeling Approach," *Doctoral Dissertation,* vol. Philosophy, Reno, University of Nevada, Reno, 2003.

[44] Dale Purves, George J. Augustine, David Fitzpatrick, William C. Hall, Anthony-Samuel Lamantia and Leonard E. White, Ed., *Neuroscience.* Sinauer Associates, 2012.

[45] Bard Ermentrout and David Terman, Ed., *Foundations of Mathematical Neuroscience.* 2010.

[46] Brayan Kolb and Ian Q. Whishaw, Ed., *An Introduction to Brain and Behavior.* Worth Publishers, 2006.

[47] A. V. M. Herz, T. Gollisch, C. K. Machens and D. Jaeger, "Modeling single-neuron dynamics and computations: a balance of detail and abstraction," *Science,* vol. 314, pp. 80, 2006.

[48] W. RALL, "Branching dendritic trees and motoneuron membrane resistivity," *Exp. Neurol.,* vol. 1, pp. 491-527, Nov, 1959.

[49] D. Jaeger, E. De Schutter and J. M. Bower, "The role of synaptic and voltage-gated currents in the control of Purkinje cell spiking: a modeling study," *J. Neurosci.,* vol. 17, pp. 91-106, Jan 1, 1997.

[50] A. Destexhe, M. Neubig, D. Ulrich and J. Huguenard, "Dendritic low-threshold calcium currents in thalamic relay cells," *J. Neurosci.,* vol. 18, pp. 3574-3588, May 15, 1998.

[51] C. C. McIntyre, W. M. Grill, D. L. Sherman and N. V. Thakor, "Cellular effects of deep brain stimulation: model-based analysis of activation and inhibition," *J. Neurophysiol.,* vol. 91, pp. 1457-1469, Apr, 2004.

[52] R. D. Traub, D. Contreras, M. O. Cunningham, H. Murray, F. E. LeBeau, A. Roopun, A. Bibbig, W. B. Wilent, M. J. Higley and M. A. Whittington, "Single-column

thalamocortical network model exhibiting gamma oscillations, sleep spindles, and epileptogenic bursts," *J. Neurophysiol.,* vol. 93, pp. 2194-2232, Apr, 2005.

[53] A. Kepecs, X. J. Wang and J. Lisman, "Bursting neurons signal input slope," *J. Neurosci.,* vol. 22, pp. 9053-9062, Oct 15, 2002.

[54] M. Stemmler and C. Koch, "How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate," *Nat. Neurosci.,* vol. 2, pp. 521-527, Jun, 1999.

[55] A. L. HODGKIN and A. F. HUXLEY, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.,* vol. 117, pp. 500-544, Aug, 1952.

[56] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Trans. Neural Netw.,* vol. 15, pp. 1063-1070, Sep, 2004.

[57] E. J. Chichilnisky, "A simple white noise analysis of neuronal light responses," *Network,* vol. 12, pp. 199-213, May, 2001.

[58] Liam Paninski, "*Estimation of Entropy and Mutual Information* ," *Neural Computation,* vol. 15, pp. 1191-1253, 2003.

[59] T. O. Sharpee, H. Sugihara, A. V. Kurgansky, S. P. Rebrik, M. P. Stryker and K. D. Miller, "Adaptive filtering enhances information transmission in visual cortex," *Nature,* vol. 439, pp. 936-942, Feb 23, 2006.

[60] M. J. Berry 2nd, I. H. Brivanlou, T. A. Jordan and M. Meister, "Anticipation of moving stimuli by the retina," *Nature,* vol. 398, pp. 334-338, Mar 25, 1999.

[61] Wulfram Gerstner and Romain Brette, "Adaptive Exponential Integrate and Fire Model," *Scholarpedia,* 2009.

[62] Fitzhugh –Nagumo (FN) Model, "http://en.wikipedia.org/wiki/FitzHugh%E2%80%93Nagumo_model;" .

[63] R. T. Faghih, K. Savla, M. A. Dahleh and E. N. Brown, "Broad range of neural dynamics from a time-varying FitzHugh-Nagumo model and its spiking threshold estimation," *IEEE Trans. Biomed. Eng.,* vol. 59, pp. 816-823, Mar, 2012.

[64] Harold Lecar, "Morris-Lecar Model," *Scholarpedia,* 2007.

[65] Q. J. Huys, M. B. Ahrens and L. Paninski, "Efficient estimation of detailed single-neuron models," *J. Neurophysiol.,* vol. 96, pp. 872-890, Aug, 2006.

[66] A. H. Jazwinski, Ed., *Stochastic Processes and Filtering Theory.* Elsivier, 1970.

[67] Simon Haykin, Ed., *Kalman Filtering and Neural Networks.* John Wiley & Sons, Inc., 2001.

[68] A. E. Gelb, Ed., *Applied Optimal Estimation.* MIT Press, 1974.

[69] S-V. I. Rosti and M. J. F. Gales, "Generalized linear gaussian models," Cambridge University, Tech. Rep. TR.420, 2001.

[70] RE Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering,* 1960.

[71] S. J. Julier and J. K. Uhlmann, "A New Extension of Kalman Filter to Nonlinear Systems," *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls 3. Retrieved 2008-05-03,* 1997.

[72] Julier, S. J., and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of IEEE,* pp. 410-422, 2004.

[73] S. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," *Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford, OC1 3PJ United Kingdom, Tech.Rep,* 1996.

[74] Akaike H, "Maximum likelihood identification of Gaussian autoregressive moving average models," *Biometrica,* 1973.

[75] Joseph J. LaViola Jr, "A Comparison of Unscented and Extended Kalman Filtering for Estimating Quaternion Motion," *IEEE American Control Society,* 2003.

[76] van der Merwe, "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," *Ph.D. Thesis, OGI Sch. of Sci. and Eng., Oreg. Health and Sci. Univ., Portland, Oreg.,* 2004.

[77] Song, Q. & Han, J. D., "A novel adaptive unscented Kalman filter for nonlinear estimation," *Proceedings of the 46th IEEE Conference on Decision and Control,* 2007.

[78] Lankarany, Milad., Zhu, Wei Ping., Swamy, M. N. S., "Joint Estimation of States and Parameters of Hodgkin-Huxley Model using Kalman Filtering," *Neurocomputing,* Under 2nd Review Process .

[79] M. Lankarany, W. P. Zhu and and Swamy, M. N. S, "Parameter estimation of hodgkin-huxley neuronal model using dual extended kalman filtering," in *IEEE International Symposium on Circuits and Systems (ISCAS),* China, 2013, .

[80] B. Fontaine, V. Benichoux, P. X. Joris and R. Brette, "Predicting spike timing in highly synchronous auditory neurons at different sound levels," *J. Neurophysiol.,* Jul 17, 2013.

[81] W. Gerstner and R. Naud, "Neuroscience. How good are neuron models?" *Science,* vol. 326, pp. 379-380, Oct 16, 2009.

[82] R. F. Fox, "Stochastic versions of the Hodgkin-Huxley equations," *Biophys. J.,* vol. 72, pp. 2068-2074, May, 1997.

[83] R. Jolivet, R. Kobayashi, A. Rauch, R. Naud, S. Shinomoto and W. Gerstner, "A benchmark test for a quantitative assessment of simple neuron models," *J. Neurosci. Methods,* vol. 169, pp. 417-424, Apr 30, 2008.

[84] http://www.incf.org/community/competitions/spike-time-prediction/2009/challenge-a," .

[85] R. Brette, Z. Piwkowska, C. Monier, M. Rudolph-Lilith, J. Fournier, M. Levy, Y. Fregnac, T. Bal and A. Destexhe, "High-resolution intracellular recordings using a real-time computational model of the electrode," *Neuron,* vol. 59, pp. 379-391, Aug 14, 2008.

[86] L. Badel, S. Lefort, T. K. Berger, C. C. Petersen, W. Gerstner and M. J. Richardson, "Extracting non-linear integrate-and-fire models from experimental data using dynamic I-V curves," *Biol. Cybern.,* vol. 99, pp. 361-370, Nov, 2008.

[87] M. Wehr and A. M. Zador, "Balanced inhibition underlies tuning and sharpens spike timing in auditory cortex," *Nature,* vol. 426, pp. 442-446, Nov 27, 2003.

[88] Y. Shu, A. Hasenstaub and D. A. McCormick, "Turning on and off recurrent balanced cortical activity," *Nature,* vol. 423, pp. 288-293, May 15, 2003.

[89] M. R. DeWeese and A. M. Zador, "Non-Gaussian membrane potential dynamics imply sparse, synchronous activity in auditory cortex," *J. Neurosci.,* vol. 26, pp. 12206-12218, Nov 22, 2006.

[90] G. J. Murphy and F. Rieke, "Network variability limits stimulus-evoked spike timing precision in retinal ganglion cells," *Neuron,* vol. 52, pp. 511-524, Nov 9, 2006.

[91] B. Haider, M. Hausser and M. Carandini, "Inhibition dominates sensory responses in the awake cortex," *Nature,* vol. 493, pp. 97-100, Jan 3, 2013.

[92] A. Destexhe and D. Contreras, "Neuronal computations with stochastic network states," *Science,* vol. 314, pp. 85-90, Oct 6, 2006.

[93] L. Paninski, M. Vidne, B. DePasquale and D. G. Ferreira, "Inferring synaptic inputs given a noisy voltage trace via sequential Monte Carlo methods," *J. Comput. Neurosci.,* vol. 33, pp. 1-19, Aug, 2012.

[94] M. Lankarany, W. P. Zhu, M. N. Swamy and T. Toyoizumi, "Inferring trial-to-trial excitatory and inhibitory synaptic inputs from membrane potential using Gaussian mixture Kalman filtering," *Front. Comput. Neurosci.,* vol. 7, pp. 109, Sep 3, 2013.

[95] A. Y. Tan, L. I. Zhang, M. M. Merzenich and C. E. Schreiner, "Tone-evoked excitatory and inhibitory synaptic conductances of primary auditory cortex neurons," *J. Neurophysiol.,* vol. 92, pp. 630-643, Jul, 2004.

[96] Jayesh H. Kotecha and Peter M. Djuric, "Gaussian Sum Particle Filtering," *IEEE Transactions on Signal Processing,* vol. 51, 2003.

[97] Nachi Gupta and Raphael Hauser, "Kalman filtering with equality and inequality state constraints," Oxford University Computing Laboratory, Tech. Rep. 07/18, 2007.

[98] I. C. CVX Research, "Matlab software for disciplined convex programming," *Version 2. 0 Beta. Http://cvxr. com/cvx,* Sep 2012.

[99] S. Song, P. J. Sjostrom, M. Reigl, S. Nelson and D. B. Chklovskii, "Highly nonrandom features of synaptic connectivity in local cortical circuits," *PLoS Biol.,* vol. 3, pp. e68, Mar, 2005.

[100] S. Lefort, C. Tomm, J. C. Floyd Sarria and C. C. Petersen, "The excitatory neuronal network of the C2 barrel column in mouse primary somatosensory cortex," *Neuron,* vol. 61, pp. 301-316, Jan 29, 2009.

[101] Y. Ikegaya, T. Sasaki, D. Ishikawa, N. Honma, K. Tao, N. Takahashi, G. Minamisawa, S. Ujita and N. Matsuki, "Interpyramid spike transmission stabilizes the sparseness of recurrent network activity," *Cereb. Cortex,* vol. 23, pp. 293-304, Feb, 2013.

[102] J. Cafaro and F. Rieke, "Noise correlations improve response fidelity and stimulus encoding," *Nature,* vol. 468, pp. 964-967, Dec 16, 2010.

[103] C. Bedard, S. Behuret, C. Deleuze, T. Bal and A. Destexhe, "Oversampling method to extract excitatory and inhibitory conductances from single-trial membrane potential recordings," *J. Neurosci. Methods,* vol. 210, pp. 3-14, Sep 15, 2012.

[104] Simon Haykin, "AdaptiveFilter Theory," *Prentice Hall,* 2002.

[105] S. H. Abadi, D. Rouseff and D. R. Dowling, "Blind deconvolution for robust signal estimation and approximate source localization," *J. Acoust. Soc. Am.,* vol. 131, pp. 2599-2610, Apr, 2012.

[106] S. H. Abadi, H. C. Song and D. R. Dowling, "Broadband sparse-array blind deconvolution using frequency-difference beamforming," *J. Acoust. Soc. Am.,* vol. 132, pp. 3018-3029, Nov, 2012.

[107] H. Liu, T. Zhang, L. Yan, H. Fang and Y. Chang, "A MAP-based algorithm for spectroscopic semi-blind deconvolution," *Analyst,* vol. 137, pp. 3862-3873, Aug 21, 2012.

[108] M. Lankarany, W. P. Zhu, M. N. S. Swamy and T. Toyoizumi, "Blind deconvolution of hodgkin-huxley model," in *35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS),* Osaka, Japan, 2013, .

[109] I. Arasaratnam and S. Haykin, "Cubature Kalman Filters," *IEEE Transactions on Automatic Control,* vol. 54, 2009.

[110] J. Dunik, M. Simandl and O. Straka, "Unscented Kalman Filter: Aspects and Adaptive Setting of Scaling Parameter," *IEEE Transactions on Automatic Control,* vol. 57, 2012.

Appendix A

Derivation of Transition Functions in (3.30) and (3.36)

First, we calculate the components of matrix $\dot{F}_x[\boldsymbol{x}(k), \boldsymbol{w}(k)]$ as follows.

$$\dot{F}_x[\boldsymbol{x}(k), \boldsymbol{w}(k)] = \begin{bmatrix} \frac{\partial F_1}{\partial V}, \frac{\partial F_1}{\partial n}, \frac{\partial F_1}{\partial m}, \frac{\partial F_1}{\partial h} \\ \frac{\partial F_2}{\partial V}, \frac{\partial F_2}{\partial n}, \frac{\partial F_2}{\partial m}, \frac{\partial F_2}{\partial h} \\ \frac{\partial F_3}{\partial V}, \frac{\partial F_3}{\partial n}, \frac{\partial F_3}{\partial m}, \frac{\partial F_3}{\partial h} \\ \frac{\partial F_4}{\partial V}, \frac{\partial F_4}{\partial n}, \frac{\partial F_4}{\partial m}, \frac{\partial F_4}{\partial h} \end{bmatrix} \tag{A.1}$$

It is required to calculate the partial derivatives of F_1, F_2, F_3 and F_4 with respect to V, n, m and h, while assuming $\boldsymbol{w}$ to be a constant. Using (3.2), the first column is thus given by

$$\begin{aligned} \frac{\partial F_1}{\partial V} &= \left[- g_{Na} m^3 h - g_K n^4 - g_L \right] / C_M \\ \frac{\partial F_2}{\partial V} &= \frac{d\alpha_n(V)}{dV}(1-n) - \frac{d\beta_n(V)}{dV} n \\ \frac{\partial F_3}{\partial V} &= \frac{d\alpha_m(V)}{dV}(1-m) - \frac{d\beta_m(V)}{dV} m \\ \frac{\partial F_4}{\partial V} &= \frac{d\alpha_h(V)}{dV}(1-h) - \frac{d\beta_h(V)}{dV} h \end{aligned} \tag{A.2}$$

The second column is obtained as

$$\frac{\partial F_1}{\partial n} = -4g_K n^3 (V - E_K)/C_M$$
$$\frac{\partial F_2}{\partial n} = -\alpha_n(V) - \beta_n(V) \tag{A.3}$$
$$\frac{\partial F_3}{\partial n} = \frac{\partial F_4}{\partial n} = 0$$

The third column can be expressed as

$$\frac{\partial F_1}{\partial m} = -3g_{Na} m^2 h(V - E_{Na})/C_M$$
$$\frac{\partial F_3}{\partial m} = -\alpha_m(V) - \beta_m(V) \tag{A.4}$$
$$\frac{\partial F_2}{\partial m} = \frac{\partial F_4}{\partial m} = 0$$

And finally, the fourth column is given by

$$\frac{\partial F_1}{\partial h} = -g_{Na} m^3 (V - E_{Na})/C_M$$
$$\frac{\partial F_4}{\partial h} = -\alpha_h(V) - \beta_h(V) \tag{A.5}$$
$$\frac{\partial F_2}{\partial h} = \frac{\partial F_3}{\partial h} = 0$$

Furthermore, we have the following derivatives:

$$\frac{d\alpha_n(V)}{dV} = \frac{0.01\left(1 - \exp\left\{-(V+55)/10\right\}\right) - 0.001\exp\left\{-(V+55)/10\right\}(V+55)}{\left(1 - \exp\left\{-(V+55)/10\right\}\right)^2}$$

$$\frac{d\alpha_m(V)}{dV} = \frac{0.1\left(1 - \exp\left\{-(V+40)/10\right\}\right) - 0.01\exp\left\{-(V+40)/10\right\}(V+40)}{\left(1 - \exp\left\{-(V+40)/10\right\}\right)^2}$$

$$\frac{d\alpha_h(V)}{dV} = -\frac{0.07}{20}\exp\left\{-(V+65)/20\right\}$$

$$\frac{d\beta_n(V)}{dV} = -\frac{0.125}{80}\exp\left\{-(V+65)/80\right\}$$

$$\frac{d\beta_m(V)}{dV} = -\frac{4}{18}\exp\{-(V+65)/80\}$$

$$\frac{d\beta_h(V)}{dV} = \frac{0.1\exp\{-(V+35)/10\}}{(1+\exp\{-(V+35)/10\})^2}$$

In order to calculate $\dot{F}_{\boldsymbol{w}}[\boldsymbol{x}(k),\boldsymbol{w}(k)]$ in (3.30), it is required to calculate the partial derivatives of F_1, F_2, F_3 and F_4 with respect to g_{Na}, g_K and g_L, while assuming $\boldsymbol{x}$ to be constant. One can easily see that F_2, F_3, and F_4 have no derivatives with respect to these parameters. Thus $\dot{F}_{\boldsymbol{w}}[\boldsymbol{x}(k),\boldsymbol{w}(k)]$ can be simplified as

$$\dot{F}_{\boldsymbol{w}}[\boldsymbol{x}(k),\boldsymbol{w}(k)] = \begin{bmatrix} \frac{\partial F_1}{\partial g_{Na}}, \frac{\partial F_1}{\partial g_K}, \frac{\partial F_1}{\partial g_L} \\ \boldsymbol{0}_{3\times 3} \end{bmatrix} \tag{A.6}$$

where, the first row can be computed as follows:

$$\begin{aligned} \frac{\partial F_1}{\partial g_{Na}} &= \frac{-m^3 h(V-E_{Na})}{C_M} \\ \frac{\partial F_1}{\partial g_K} &= \frac{-n^4 h(V-E_K)}{C_M} \\ \frac{\partial F_1}{\partial g_L} &= \frac{-(V-E_L)}{C_M} \end{aligned} \tag{A.7}$$

Now, we want to calculate the transition function $\dot{\Gamma}_{\boldsymbol{x}}[\boldsymbol{x}(k),\boldsymbol{w}(k)]$ in (3.36). Similar to (A.1), we can write the components of this transition function as follows.

$$\dot{\Gamma}_x[\boldsymbol{x}(k), \boldsymbol{w}(k)] = \begin{bmatrix} \frac{\partial \Gamma_1}{\partial V}, \frac{\partial \Gamma_1}{\partial n}, \frac{\partial \Gamma_1}{\partial m}, \frac{\partial \Gamma_1}{\partial h} \\ \frac{\partial \Gamma_2}{\partial V}, \frac{\partial \Gamma_2}{\partial n}, \frac{\partial \Gamma_2}{\partial m}, \frac{\partial \Gamma_2}{\partial h} \\ \frac{\partial \Gamma_3}{\partial V}, \frac{\partial \Gamma_3}{\partial n}, \frac{\partial \Gamma_3}{\partial m}, \frac{\partial \Gamma_3}{\partial h} \\ \frac{\partial \Gamma_4}{\partial V}, \frac{\partial \Gamma_4}{\partial n}, \frac{\partial \Gamma_4}{\partial m}, \frac{\partial \Gamma_4}{\partial h} \end{bmatrix} \tag{A.8}$$

where the partial derivatives of Γ_1, Γ_2, Γ_3 and Γ_4 with respect to V are given by

$$\begin{aligned} \frac{\partial \Gamma_1}{\partial V} &= \left(- g_{Na} m^3 h - g_K n^4 - g_L\right) / C_M \\ \frac{\partial \Gamma_2}{\partial V} &= \left[\frac{dn_\infty(V)}{dV}\tau_n(V) - \frac{d\tau_n(V)}{dV}\left(n_\infty(V) - n\right)\right] \Big/ \left[\tau_n(V)\right]^2 \\ \frac{\partial \Gamma_3}{\partial V} &= \left[\frac{dm_\infty(V)}{dV}\tau_m(V) - \frac{d\tau_m(V)}{dV}\left(m_\infty(V) - m\right)\right] \Big/ \left[\tau_m(V)\right]^2 \\ \frac{\partial \Gamma_4}{\partial V} &= \left[\frac{dh_\infty(V)}{dV}\tau_h(V) - \frac{d\tau_h(V)}{dV}\left(h_\infty(V) - h\right)\right] \Big/ \left[\tau_h(V)\right]^2 \end{aligned} \tag{A.9}$$

The second column of (A.8) is obtained as

$$\begin{aligned} \frac{\partial \Gamma_1}{\partial n} &= -4 g_K n^3 (V - E_K) / C_M \\ \frac{\partial \Gamma_2}{\partial n} &= -1 / \tau_n(V) \\ \frac{\partial \Gamma_3}{\partial n} &= \frac{\partial \Gamma_4}{\partial n} = 0 \end{aligned} \tag{A.10}$$

The third column of (A.8) can be expressed as

$$\frac{\partial \Gamma_1}{\partial m} = -3 g_{Na} m^2 h (V - E_{Na}) / C_M$$
$$\frac{\partial \Gamma_3}{\partial m} = -1 / \tau_m(V) \qquad \text{(A.11)}$$
$$\frac{\partial \Gamma_2}{\partial m} = \frac{\partial \Gamma_4}{\partial m} = 0$$

And finally, the fourth column of (A.8) is given by

$$\frac{\partial \Gamma_1}{\partial h} = -g_{Na} m^3 (V - E_{Na}) / C_M$$
$$\frac{\partial \Gamma_4}{\partial h} = -1 / \tau_h(V) \qquad \text{(A.12)}$$
$$\frac{\partial \Gamma_2}{\partial h} = \frac{\partial \Gamma_3}{\partial h} = 0$$

Furthermore, we have the following derivatives for q= *n, m, h*:

$$\frac{dq_\infty(V)}{dV} = s_q \exp\left\{-s_q\left(V - V_{th}(q)\right)\right\} q_\infty{}^2$$
$$\frac{d\tau_q(V)}{dV} = -\frac{s_q}{2t_q} \sinh\left\{\frac{s_q\left(V - V_{th}(q)\right)}{2}\right\} \tau_q{}^2$$

In order to calculate $\dot{\Gamma}_{\boldsymbol{w}}[\boldsymbol{x}(k), \boldsymbol{w}(k)]$ in (3.36), it is required to calculate the partial derivatives of F_1, F_2, F_3 and F_4 with respect to $\boldsymbol{w}$ = [g_{Na}, g_K, g_L, E_{Na}, E_K, E_L, $V_{th(n)}$, $V_{th(m)}$, $V_{th(h)}$, s_n, s_m, s_h, t_n, t_m, t_h], while assuming $\boldsymbol{x}$ to be constant. Thus $\dot{\Gamma}_{\boldsymbol{w}}[\boldsymbol{x}(k), \boldsymbol{w}(k)]$ can be written as

$$\dot{\Gamma}_{\boldsymbol{w}}[\boldsymbol{x}(k),\boldsymbol{w}(k)]=\begin{bmatrix} \dfrac{\partial\Gamma_1}{\partial g_{Na}},\dfrac{\partial\Gamma_1}{\partial g_K},\dfrac{\partial\Gamma_1}{\partial g_L}\dfrac{\partial\Gamma_1}{\partial E_{Na}},\dfrac{\partial\Gamma_1}{\partial E_K},\dfrac{\partial\Gamma_1}{\partial E_L},\ \boldsymbol{0}_{1\times 9} \\ \boldsymbol{0}_{1\times 6},\ \dfrac{\partial\Gamma_2}{\partial V_{th}(n)},\dfrac{\partial\Gamma_2}{\partial s_n},\dfrac{\partial\Gamma_2}{\partial t_n},\ \boldsymbol{0}_{1\times 6} \\ \boldsymbol{0}_{1\times 9},\ \dfrac{\partial\Gamma_3}{\partial V_{th}(m)},\dfrac{\partial\Gamma_3}{\partial s_m},\dfrac{\partial\Gamma_3}{\partial t_m},\ \boldsymbol{0}_{1\times 3} \\ \boldsymbol{0}_{1\times 12},\ \dfrac{\partial\Gamma_4}{\partial V_{th}(h)},\dfrac{\partial\Gamma_4}{\partial s_h},\dfrac{\partial\Gamma_4}{\partial t_h} \end{bmatrix} \tag{A.13}$$

where the components of the first row are calculated as follows.

$$\begin{aligned} \frac{\partial\Gamma_1}{\partial g_{Na}} &= \frac{-m^3h(V-E_{Na})}{C_M} \\ \frac{\partial\Gamma_1}{\partial g_K} &= \frac{-n^4h(V-E_K)}{C_M} \\ \frac{\partial\Gamma_1}{\partial g_L} &= \frac{-(V-E_L)}{C_M} \\ \frac{\partial\Gamma_1}{\partial E_{Na}} &= \frac{g_{Na}m^3h}{C_M} \\ \frac{\partial\Gamma_1}{\partial E_K} &= \frac{g_Kn^4}{C_M} \\ \frac{\partial\Gamma_1}{\partial E_L} &= \frac{g_L}{C_M} \end{aligned} \tag{A.14}$$

The components of the other rows can be written in a general form for, respectively, i = *1, 2, 3* and q = *n, m, h*.

$$\frac{\partial \Gamma_i}{\partial V_{th}(q)} = \left[\frac{\partial q_\infty}{\partial V_{th}(q)} \tau_q - \frac{\partial \tau_q}{\partial V_{th}(q)} \left(q_\infty(V) - q \right) \right] \Big/ \tau_q{}^2$$

$$\frac{\partial \Gamma_i}{\partial s_q} = \left[\frac{\partial q_\infty}{\partial s_q} \tau_q - \frac{\partial \tau_q}{\partial s_q} \left(q_\infty(V) - q \right) \right] \Big/ \tau_q{}^2 \qquad \text{(A.15)}$$

$$\frac{\partial \Gamma_i}{\partial t_q} = \left[\frac{\partial q_\infty}{\partial t_q} \tau_q - \frac{\partial \tau_q}{\partial t_q} \left(q_\infty(V) - q \right) \right] \Big/ \tau_q{}^2$$

where,

$$\frac{\partial q_\infty}{\partial V_{th}(q)} = -s_q \exp\left\{ -s_q \left(V - V_{th}(q) \right) \right\} q_\infty{}^2, \quad \frac{\partial \tau_q}{\partial V_{th}(q)} = \frac{s_q}{2t_q} \sinh\left\{ \frac{s_q \left(V - V_{th}(q) \right)}{2} \right\} \tau_q{}^2$$

$$\frac{\partial q_\infty}{\partial s_q} = \left(V - V_{th}(q) \right) \exp\left\{ -s_q \left(V - V_{th}(q) \right) \right\} q_\infty{}^2, \quad \frac{\partial \tau_q}{\partial s_q} = \frac{\left(V - V_{th}(q) \right)}{2t_q} \sinh\left\{ \frac{s_q \left(V - V_{th}(q) \right)}{2} \right\} \tau_q{}^2$$

$$\frac{\partial q_\infty}{\partial t_q} = 0 \ , \quad \frac{\partial \tau_q}{\partial t_q} = \cosh^{-1}\left\{ \frac{s_q \left(V - V_{th}(q) \right)}{2} \right\}$$

Appendix B

Hints to proof (4.5), (4.6) and (4.7)

Hints to Proof (4.6)

As mentioned in Chapter 4, the excitatory/inhibitory synaptic conductances g_E and g_I, are stochastic variables changing over trials. According to our assumptions in subsection 4.3.1, g_E and g_I, for each time t, can be written as:

$$g_E = \bar{g}_E + \Delta g_E \xi_E \ , \quad g_I = \bar{g}_I + \Delta g_I \xi_I \tag{B.1}$$

where Δg_E and Δg_I are respectively the amplitudes of variations in excitatory and inhibitory synaptic conductances, ξ_E and ξ_I are the corresponding stochastic parts modeled by white Gaussian noise of zero mean and unit variance, and $\bar{g}_E$ and $\bar{g}_I$ are the trial means of excitatory and inhibitory synaptic conductances, respectively. Moreover, the membrane potential of each trial can also be expressed by

$$V = \bar{V} + \Delta V \, \xi_V \tag{B.2}$$

where $\bar{V}$, ΔV and ξ_V are the trial mean, the amplitude of variation and the stochastic part of the membrane potential, respectively. It is important to note that the stochastic part of the membrane potential is correlated with that of the excitatory and inhibitory synaptic conductances.

Recalling (4.3), and averaging over different trials, we have

$$C_M \frac{d\overline{V}}{dt} = -\overline{g_L(V - V_L)} - \overline{g_E(V - V_E)} - \overline{g_I(V - V_I)} + \bar{I} \tag{B.3}$$

Using (B.1) and (B.2) in (B.3), and accomplishing some calculus using *Mathematica*, (B.3) is stated as follows.

$$\begin{aligned} C_M \frac{d\overline{V}}{dt} &= -g_L(\overline{V} - V_L) - \left(\overline{g_E V} - \bar{g}_E V_E\right) - \left(\overline{g_I V} - \bar{g}_I V_I\right) + \bar{I} \\ &= -g_L(\overline{V} - V_L) - \left(\bar{g}_E \overline{V} + \Delta g_E \Delta V \overline{\xi_E \xi_V} - \bar{g}_E V_E\right) - \left(\bar{g}_I \overline{V} + \Delta g_I \Delta V \overline{\xi_I \xi_V} - \bar{g}_I V_I\right) + \bar{I} \end{aligned} \tag{B.4}$$

Now, by using (B.4) and equating it by (4.4), we can easily infer $\hat{g}_E$ and $\hat{g}_I$ from (4.5).

Hints to Proof (4.6)

As mentioned in subsection 4.3.1, if the excitatory and inhibitory synaptic conductances are independent and changing much slower than the dynamics of the membrane potential, we can assume that V immediately converges to an adiabatic fixed point defined by the quasi-stationary values of g_E and g_I, namely,

$$V = \frac{I + g_L V_L + g_E V_E + g_I V_I}{g_E + g_I + g_L} \tag{B.5}$$

Now, assuming that the injected current is the same for different trials, by averaging over trials we will have:

$$\begin{aligned} \overline{V} &= \overline{\left(\frac{I + g_L V_L + g_E V_E + g_I V_I}{g_E + g_I + g_L}\right)} \\ &= V_0 + I / g \end{aligned} \tag{B.6}$$

where,

$$V_0 = \overline{\left(\frac{g_L V_L + g_E V_E + g_I V_I}{g_E + g_I + g_L} \right)}$$
$$g = \overline{\left(\frac{1}{g_E + g_I + g_L} \right)} \tag{B.7}$$

By LS method, we aim to find $\hat{g}_E$ and $\hat{g}_I$ such that,

$$\begin{aligned} \overline{V} &= \frac{I + g_L V_L + \hat{g}_E V_E + \hat{g}_I V_I}{\hat{g}_E + \hat{g}_I + g_L} \\ &= \frac{I}{\hat{g}_E + \hat{g}_I + g_L} + \frac{g_L V_L + \hat{g}_E V_E + \hat{g}_I V_I}{\hat{g}_E + \hat{g}_I + g_L} \end{aligned} \tag{B.8}$$

By equation (B.8) to (B.6), and using (B.7), $\hat{g}_E$ and $\hat{g}_I$ can be calculated as follows.

$$\begin{aligned} \hat{g}_E &= \frac{g(V_0 - V_I) + g_L(V_I - V_L)}{V_E - V_I} \\ \hat{g}_I &= \frac{g(V_E - V_0) - g_L(V_E - V_L)}{V_E - V_I} \end{aligned} \tag{B.9}$$

In order to prove (4.6), while considering that the trial-to-trial variations of the synaptic conductances are represented by (B.1), V_0 and g are expanded as follows.

$$1/g = \frac{1}{\bar{g}_E + \bar{g}_I + g_L} \left[1 + \frac{\Delta g_E^2 + \Delta g_I^2}{(\bar{g}_E + \bar{g}_I + g_L)^2} \right] \tag{B.10}$$

$$\begin{aligned} V_0 = \frac{\bar{g}_E V_E + \bar{g}_I V_I + g_L V_L}{\bar{g}_E + \bar{g}_I + g_L} &- \frac{\bar{g}_I (V_E - V_I) + g_L (V_E - V_L)}{(\bar{g}_E + \bar{g}_I + g_L)^3} \Delta g_E^2 \\ &+ \frac{\bar{g}_E (V_E - V_I) + g_L (V_L - V_I)}{(\bar{g}_E + \bar{g}_I + g_L)^3} \Delta g_I^2 \end{aligned} \tag{B.11}$$

Now, by replacing (B.10) and (B.11) in (B.9) and simplifying it by using *Mathematica*, we can calculate $\hat{g}_E$ and $\hat{g}_I$ in (4.6).

Hints to Proof (4.7)

In order to prove the inequality used in (4.7), we can apply Jansen's inequality. We want to show that

$$E\left[\frac{x}{a+x}\right] < E[x]E\left[\frac{1}{a+x}\right],\ x>0 \tag{B.12}$$

We can write

$$\begin{aligned}
\frac{1}{E[x]}E\left[\frac{1}{a+x}\right] &= \frac{1}{E[x]}E\left[1-\frac{a}{a+x}\right] \\
&< \frac{1}{E[x]}\left(1-\frac{a}{a+E[x]}\right) \\
&= \frac{1}{E[x]}\left(\frac{E[x]}{a+E[x]}\right) \\
&= \frac{1}{a+E[x]} \\
&< E\left[\frac{1}{a+x}\right]
\end{aligned} \tag{B.13}$$

Appendix C

Number of Kalman Filters in the GMKF Algorithm

In order to show how the number of mixture Kalman filters grows exponentially with the arrival of a new observation $y(t)$, we assume that $p(\boldsymbol{x}(t\text{-}1)|y(0{:}t\text{-}1))$ can be represented by K Kalman filters [96] as follows.

$$p(\boldsymbol{x}(t-1)\,|\,y(0:t-1)) = \sum_{i=1}^{K} \beta_i(t-1)\, p(\boldsymbol{x}(t-1)\,|\,y(0:t-1), i) \tag{C.1}$$

where $\beta_i(t\text{-}1)$ is the normalized weight for the i^{th} Kalman filter at time $t\text{-}1$. Then, $p(\boldsymbol{x}(t)|y(0{:}t\text{-}1))$ will be given by the following equation considering that $p(\boldsymbol{x}(t)|\boldsymbol{x}(t\text{-}1))$ is approximated by G mixands.

$$p(\boldsymbol{x}(t)\,|\,y(0:t-1)) = \sum_{j=1}^{G} \sum_{i=1}^{K} \alpha_j \beta_i(t-1)\, p(\boldsymbol{x}(t)\,|\,y(0:t-1), i, j) \tag{C.2}$$

where $p(\boldsymbol{x}(t)|y(0{:}t\text{-}1),\ i,\ j)$ is the conditional probability of the state $\boldsymbol{x}$(t) for the i^{th} filter and j^{th} mixand given the observation up to time $t\text{-}1$. Now $p(\boldsymbol{x}(t)|y(0{:}t))$ can be calculated using the Bayesian rule.

$$
\begin{aligned}
p(\boldsymbol{x}(t) \mid y(0:t)) &= \frac{p(y(t) \mid \boldsymbol{x}(t)) p(\boldsymbol{x}(t) \mid y(0:t-1))}{p(y(t) \mid y(0:t-1))} \\
&= \frac{\sum_{j=1}^{G} \sum_{i=1}^{K} \alpha_j \beta_i(t-1) p(y(t) \mid \boldsymbol{x}(t), i, j) p(\boldsymbol{x}(t) \mid y(0:t-1), i, j)}{p(y(t) \mid y(0:t-1))} \\
&= \frac{\sum_{j=1}^{G} \sum_{i=1}^{K} \alpha_j \beta_i(t-1) p(y(t), \boldsymbol{x}(t) \mid y(0:t-1), i, j)}{p(y(t) \mid y(0:t-1))} \\
&= \frac{\sum_{j=1}^{G} \sum_{i=1}^{K} \alpha_j \beta_i(t-1) p(y(t) \mid y(0:t-1), i, j) p(\boldsymbol{x}(t) \mid y(0:t), i, j)}{p(y(t) \mid y(0:t-1))} \\
&= \sum_{j=1}^{G} \sum_{i=1}^{K} \gamma_{i,j}(t) p(\boldsymbol{x}(t) \mid y(0:t), i, j)
\end{aligned}
\tag{C.3}
$$

where

$$
\begin{aligned}
\gamma_{i,j}(t) &= p(i, j \mid y(0:t)) \\
&= \frac{\alpha_j(t) \beta_i(t-1) p(y(t) \mid y(0:t-1), i, j)}{\sum_{i=1}^{K} \sum_{j=1}^{G} \alpha_j(t) \beta_i(t-1) p(y(t) \mid y(0:t-1), i, j)}
\end{aligned}
\tag{C.4}
$$

and the denominator of (C.4), for each i and j, can be calculated as follows.

$$
\begin{aligned}
&p(y(t) \mid y(0:t-1), i, j) = \int p(y(t) \mid \boldsymbol{x}(t), i, j) p(\boldsymbol{x}(t) \mid y(0:t-1), i, j) d\boldsymbol{x}(t) \\
&= \int \int p(y(t) \mid \boldsymbol{x}(t), i, j) p(\boldsymbol{x}(t) \mid \boldsymbol{x}(t-1), j) p(\boldsymbol{x}(t-1) \mid y(0:t-1), i) d\boldsymbol{x}(t) d\boldsymbol{x}(t-1)
\end{aligned}
\tag{C.5}
$$

Although (C.5) can be easily obtained using the results of KF (see Appendix F), our main goal is to highlight that $p(\boldsymbol{x}(t)|y(0:t))$ in (C.3) is approximated by $K \times G$ filters. Consequently, $p(\boldsymbol{x}(t+1)|y(0:t+1))$ will be represented by $K \times G^2$ filters and the number of filters increases exponentially with the arrival of each new observation. As mentioned in Section III, and according to [96], this problem can be overcome by resampling. To do

this, it is required to select the most probable K filters of (C.3) from $\gamma_{i,j}(t)$. For such a case, $p(\boldsymbol{x}(t)|y(0{:}t))$ will be approximated by only K filters as follows.

$$p(\boldsymbol{x}(t) \mid y(0:t)) \approx \sum_{i=1}^{K} \beta_i(t)\, p(\boldsymbol{x}(t) \mid y(0:t), i) \tag{C.6}$$

where $\beta_i(t)$ is the weight for the K highest values of $\gamma_{i,j}(t)$. Since $\beta_i(t)$ is normalized after the resampling process, it gives equivalent weights to all the selected filters ($1/K$ for each filter). Please note that the resampling process is done only to eliminate the increase in the number of filters required for approximating $p(\boldsymbol{x}(t)|y(0{:}t))$ at the arrival of new observation $y(t)$. Hence, $K \times G$ Kalman filters are needed to run at each time t to compute $p(\boldsymbol{x}(t)|y(0{:}t))$ for each new observation $y(t)$.

Appendix D

Kalman Forward Filtering ([69])

In this appendix, the Kalman forward filtering is derived for each i belonging to $\{1{:}K\}$ and j belonging to $\{1{:}G\}$. The standard KF technique includes two main steps: time update and measurement update [67]. In the time update step, for each i and j, the predicted state estimate $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0{:}t\text{-}1)\}$ and the predicted state correlation matrix $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)\,(\boldsymbol{x}_{i,j}(t))^{H}\,|y(0{:}t\text{-}1)\}$ (therefore the state covariance matrix, $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)\,(\boldsymbol{x}_{i,j}(t))^{H}\,|y(0{:}t\text{-}1)\} - (\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0{:}t\text{-}1)\})^{2}$) are calculated using (4.1) (or (4.4), i.e., the system dynamic). Then, in the measurement update step, the updated state estimate $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0{:}t)\}$ and updated state correlation matrix $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)\,(\boldsymbol{x}_{i,j}(t))^{H}\,|y(0{:}t)\}$ are calculated using the standard Kalman filtering. In the following, for the sake of simplicity in representing the equations, $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0{:}t\text{-}1)\}$ and $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0{:}t)\}$ are denoted as $\boldsymbol{x}^{t\text{-}1}{}_{i,j}(t)$ and $\boldsymbol{x}^{t}{}_{i,j}(t)$, respectively. Accordingly, the predicted and updated state covariance matrices are denoted as $\Sigma^{t-1}_{x_{i,j}}(t)$ and $\Sigma^{t}_{x_{i,j}}(t)$, respectively. Please note that the symbol Σ (small sigma whose indices appear as subscript and superscript) that is used in the thesis to represent the covariance matrix is consistent with the reference [96]. Moreover the symbol $\sum$ (capital sigma whose range appear at the bottom and top) is used for the summation. For each i and j, the time update step can be followed by

$$\begin{aligned}\boldsymbol{x}^{t-1}_{i,j}(t) &= A(t)\boldsymbol{x}^{t-1}_{i}(t-1) + \boldsymbol{\mu}_{v_j}(t-1)\\ \Sigma^{t-1}_{x_{i,j}}(t) &= A(t)\Sigma^{t-1}_{x_i}(t-1)A(t)^{H} + \Sigma_{v_j}\end{aligned} \tag{D.1}$$

where $A(t) = \left.\frac{\partial F[x]}{\partial x}\right|_{x(t)}$ is the time dependent transition matrix, and $\boldsymbol{\mu}_{v_j}$ and Σ_{v_j} are the mean and variance of the synaptic input corresponding to j^{th} mixand. The so called Kalman gain is then obtained by

$$K_{i,j}(t) = \sum_{x_{i,j}}^{t-1}(t) C^H \sum_{y_{i,j}}^{-1}(t) \tag{D.2}$$

where

$$\sum_{y_{i,j}}(t) = C \sum_{x_{i,j}}^{t-1}(t) C^H + \sigma_{\varepsilon}^2 \tag{D.3}$$

Here, $C = [1,0,0]$ is the observation vector and σ^2_{ε} is the observation noise variance. By defining the innovation process as $e_{i,j}(t) = y(t) - C\boldsymbol{x}_{i,j}^{t-1}(t)$, the updated state estimate and state covariance matrix are calculated as follows.

$$\begin{aligned} \boldsymbol{x}_{i,j}^{t}(t) &= \boldsymbol{x}_{i,j}^{t-1}(t) + K_{i,j}(t) e_{i,j}(t) \\ \sum_{x_{i,j}}^{t}(t) &= \sum_{x_{i,j}}^{t-1}(t) - K_{i,j}(t)\, C \sum_{x_{i,j}}^{t-1}(t) \end{aligned} \tag{D.4}$$

Since the KF-based algorithm is implemented in recursive manner, (D.4) is used for the next iteration and $\boldsymbol{x}^t_{i,j}(t)$ (or $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0{:}t)\}$) is estimated for all the time samples of the observation data $y(0{:}T)$.

Appendix E

Kalman Backward Filtering (Smoothing) [69]

Similar to Appendix D, Kalman backward filtering is accomplished for $K \times G$ Kalman filters at each iteration t. The resampling procedure is already performed to eliminate the increase in the number of filters estimating $p(\boldsymbol{x}(t)|y(0:t))$. The goal with backward filtering is providing a better estimate of the state mean $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0:T)\}$ and the state correlation matrix $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)\ (\boldsymbol{x}_{i,j}(t))^H\ |y(0:T)\}$ using all the observed data $y(0:T)$. Again, for the sake of simplicity, $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0:T)\}$ and the state covariance matrix ($\mathrm{E}\{\boldsymbol{x}_{i,j}(t)\ (\boldsymbol{x}_{i,j}(t))^H |y(0:T)\} - (\mathrm{E}\{\boldsymbol{x}_{i,j}(t)|y(0:T)\})^2$) are denoted as $\hat{\boldsymbol{x}}_{i,j}^{t}(t)$ and $\hat{\Sigma}_{x_{i,j}}^{t}(t)$, respectively. Following the standard Kalman backward filtering algorithm [96], for each i belonging to $\{1:K\}$ and j belonging to $\{1:G\}$, we have:

$$\hat{\boldsymbol{x}}_{i,j}^{t-1}(t-1) = \boldsymbol{x}_{i,j}^{t-1}(t-1) + J_{i,j}(t-1)\left(\hat{\boldsymbol{x}}_{i,j}^{t}(t) - \boldsymbol{x}_{i,j}^{t-1}(t)\right) \tag{E.1}$$

where, according to [96],

$$J_{i,j}(t-1) = \Sigma_{x_{i,j}}^{t-1}(t-1)A(t)^H\left(\Sigma_{x_{i,j}}^{t-1}(t)\right)^{-1} \tag{E.2}$$

The state covariance matrix is calculated as follows (see [96]).

$$\hat{\Sigma}_{x_{i,j}}^{t-1}(t-1) = \Sigma_{x_{i,j}}^{t-1}(t-1) + J_{i,j}(t-1)\left(\hat{\Sigma}_{x_{i,j}}^{t}(t) - \Sigma_{x_{i,j}}^{t-1}(t)\right)\left(J_{i,j}(t-1)\right)^H \tag{E.3}$$

For time indices, since in backward filtering the initial values start at $t=T$, we have

$$\hat{\boldsymbol{x}}_{i,j}(T) = \boldsymbol{x}^{T}_{i,j}(T),\ \hat{\Sigma}^{T}_{x_{i,j}}(T) = \Sigma^{T}_{x_{i,j}}(T) \tag{E.4}$$

Moreover, it is necessary to obtain the correlation matrices $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)\ (\boldsymbol{x}_{i,j}(t))^{H} | y(0{:}T)\}$ and $\mathrm{E}\{\boldsymbol{x}_{i,j}(t)\ (\boldsymbol{x}_{i,j}(t\text{-}1))^{H} | y(0{:}T)\}$ for the expectation-maximization (EM) algorithm. These matrices are, respectively, denoted as $\hat{R}^{t-1}_{i,j}(t)$ and $\hat{R}^{t}_{i,j}(t)$ in our equations and can be obtained (see [96], page 47-49) by

$$\begin{aligned}
\hat{R}^{t}_{x_{i,j}}(t) &= \hat{\Sigma}^{t}_{x_{i,j}}(t) + \hat{\boldsymbol{x}}^{\boldsymbol{t}}_{\boldsymbol{i,j}}(t)\left(\hat{\boldsymbol{x}}^{\boldsymbol{t}}_{\boldsymbol{i,j}}(t)\right)^{H} \\
\hat{R}^{t-1}_{x_{i,j}}(t) &= \hat{\Sigma}^{t-1}_{x_{i,j}}(t) + \hat{\boldsymbol{x}}^{\boldsymbol{t}}_{\boldsymbol{i,j}}(t)\left(\hat{\boldsymbol{x}}^{\boldsymbol{t\text{-}1}}_{\boldsymbol{i,j}}(t-1)\right)^{H}
\end{aligned} \tag{E.5}$$

where $\hat{\Sigma}^{t-1}_{x_{i,j}}(t) = \hat{\Sigma}^{t}_{x_{i,j}}(t)\left(J_{i,j}(t-1)\right)^{H}$ [96] and for the initial values at $t = T$, we have $\hat{R}^{T}_{i,j}(T) = \hat{\Sigma}^{T}_{x_{i,j}}(T) + \hat{\boldsymbol{x}}^{\boldsymbol{T}}_{\boldsymbol{i,j}}(T)\left(\hat{\boldsymbol{x}}^{\boldsymbol{T}}_{\boldsymbol{i,j}}(T)\right)$.

Having obtained the above statistics, we can use the EM algorithm to infer the statistical parameters of each mixture.

Appendix F

Expectation-Maximization for the GMKF-Based Algorithm

Recalling (4.16) and (4.17), we want to maximize the logarithm of the joint probability of the states and observation via the EM algorithm for each mixture. Then, the results are combined to yield the final estimate of the states as well as the distributions of the synaptic inputs. Hence, we have

$$\begin{aligned}\max_{s.t.\hat{\theta}} Q(\theta,\hat{\theta}) &= \max\left(\int \sum_{i,j} p(i,j\mid Y)\log\left(p(Y,X,i,j\mid\hat{\theta})\right)p(X\mid i,j,Y)dX\right)\\ &= \max\left(E_{p(X|i,j,Y)}\left\{\sum_{i,j} p(i,j\mid Y)\log\left(p(Y,X,i,j\mid\hat{\theta})\right)\mid Y,\theta\right\}\right)\end{aligned} \tag{F.1}$$

where X and Y stand for all the states and observation over time, respectively. The expected value of the joint probability in (E.1) can be expanded as follows.

$$\begin{aligned}&E_{p(X/i,j,Y)}\left\{\sum_{i,j} p(i,j\mid Y)\log\left(p(Y,X,i,j/\hat{\theta})\right)/Y,\theta\right\} =\\ &\sum_i\sum_j\sum_{t=1}^{T} p(i,j\mid y(0:t))E_{p(X|i,j,Y)}\left\{\log\alpha_j(t)+\log\beta_i(t)+\log p(y(t)\mid x(t),i,j,\theta)+\log p(x(t)\mid x(t-1),i,j,\theta)\right\} =\\ &\sum_i\sum_j\sum_{t=1}^{T} p(i,j\mid y(0:t))E_{p(X|i,j,Y)}\left\{\log\alpha_j(t)\right\}+\sum_i\sum_j\sum_{t=1}^{T} p(i,j\mid y(0:t))E_{p(X|i,j,Y)}\left\{\log\beta_i(t)\right\}+\\ &\sum_i\sum_j\sum_{t=1}^{T} p(i,j\mid y(0:t))E_{p(X|i,j,Y)}\left\{-\frac{1}{2}\log\sigma_\varepsilon^2+\left(y(t)-x(t)\right)^H\left(\sigma_\varepsilon^2\right)^{-1}\left(y(t)-x(t)\right)\right\}+\\ &\sum_i\sum_j\sum_{t=1}^{T} p(i,j\mid y(0:t))E_{p(X|i,j,Y)}\left\{-\frac{1}{2}\log\Sigma_{v_{ij}}+\left(\boldsymbol{x}(t)-A\boldsymbol{x}(t-1)-\boldsymbol{\mu}_{v_{ij}}(t-1)\right)^H\left(\Sigma_{v_{ij}}\right)^{-1}\left(\boldsymbol{x}(t)-A\boldsymbol{x}(t-1)-\boldsymbol{\mu}_{v_{ij}}(t-1)\right)\right\}\end{aligned} \tag{F.2}$$

when using (F.2) to solve (F.1), we need to calculate $E_{p(X|i,j,Y)}\{\boldsymbol{x}(t)\}$, $E_{p(X|i,j,Y)}\{\boldsymbol{x}(t\text{-}1)(\boldsymbol{x}(t))^H\}$ and $E_{p(X|i,j,Y)}\{\boldsymbol{x}(t)\ (\boldsymbol{x}(t))^H\}$. These statistics are already calculated in the backward filtering step as indicated by $\hat{\boldsymbol{x}}_{i,j}^{t}(t)$, $\hat{R}_{i,j}^{t-1}(t)$ and $\hat{R}_{i,j}^{t}(t)$, respectively. As mentioned in Appendix C, $p(y(t)|y(0{:}t\text{-}1), i, j)$ can be simply computed by the results of the KF as follows.

$$\begin{aligned} &p(y(t)/y(0:t-1),i,j)=\int p(y(t)/\boldsymbol{x}(t),i,j)p(\boldsymbol{x}(t)/y(0:t-1),i,j)d\boldsymbol{x}(t) \\ &=\int p(y(t)/\boldsymbol{x}(t),i,j)N(\boldsymbol{x}(t);\boldsymbol{x}_{i,j}^{t-1}(t),\textstyle\sum_{\boldsymbol{x}_{i,j}}^{t-1}(t))d\boldsymbol{x}(t) \\ &=N(y^t;C\boldsymbol{x}_{i,j}^{t}(t),\sigma_\varepsilon^2+C\textstyle\sum_{\boldsymbol{x}_{i,j}}^{t}(t)C^H) \end{aligned} \tag{F.3}$$

where to calculate the second line of (F.3), the results of Kalman forward filtering are used. It is worth mentioning that computation of the integral in (F.3) is trivial since $p(y(t)|\boldsymbol{x}(t), i, j)$ and $p(\boldsymbol{x}(t)|y(0{:}t\text{-}1), i, j)$ are expressed by Gaussian distribution. Using (F.3), $p(i,j|y(0{:}t))$ can be estimated as

$$\begin{aligned} p(i,j\mid y(0:t))&=\gamma_{i,j}(t) \\ &=\frac{\alpha_j(t)\beta_i(t-1)p(y(t)\mid y(0:t-1),i,j)}{\sum_{i=1}^{K}\sum_{j=1}^{G}\alpha_j(t)\beta_i(t-1)p(y(t)\mid y(0:t-1),i,j)} \end{aligned} \tag{F.4}$$

Note that $\beta_i(t\text{-}1)$ is the normalized weight after resampling and is equal to $1/K$ for all t. Now we can derive the M step by taking the derivative of (F.2) with respect to the unknown parameters $[\alpha, \boldsymbol{\mu}_v, \Sigma_v, \sigma^2_\varepsilon]$. It should be noted that it is convenient to demonstrate the optimized parameters in the EM algorithm as $[\hat{\alpha}, \hat{\boldsymbol{\mu}}_v, \hat{\Sigma}_v, \hat{\sigma}^2_\varepsilon]$. For α_j, one can write

$$\max_{s.t.\hat{\alpha}} Q(\theta,\hat{\theta}) = \max\left(\sum_{i,j}\sum_{t=1}^{T} \gamma_{i,j}(t) E_{p(X|i,j,Y)}\left\{\log \alpha_j(t)\right\}\right) \tag{F.5}$$

and noting that $\sum_j \alpha_j = 1$. Using Lagrange multiplier, we can solve (F.5) as follows.

$$\sum_{i,j}\sum_{t=1}^{T} \gamma_{i,j}^{t} \log \alpha_j - \lambda\left(\sum_j \alpha_j - 1\right) \tag{F.6}$$

which results in solving the following set of equations.

$$\begin{cases} \sum_{i=1}^{K}\sum_{t=1}^{T} \gamma_{i,j}(t)\frac{1}{\hat{\alpha}_j} - \lambda = 0 \\ \sum_j \hat{\alpha}_j - 1 = 0 \end{cases} \Rightarrow \hat{\alpha}_j = \frac{1}{T}\sum_{i=1}^{K}\sum_{t=1}^{T}\gamma_{i,j}(t) \tag{F.7}$$

The optimized $\hat{\mu}_{\boldsymbol{v}_{i,j}}$ can be calculated by taking the derivative of (F.2) to $\hat{\mu}_{\boldsymbol{v}}$, as given below:

$$\begin{aligned} \frac{\partial Q(\theta,\hat{\theta})}{\partial \hat{\mu}_v} &= \frac{\partial\left(\sum_{i,j}\sum_{t=1}^{T} \gamma_{i,j}(t) E_{p(X|i,j,Y)}\left\{-\frac{1}{2}\log\Sigma_{v_{i,j}} + \left(\boldsymbol{x}(t) - A\boldsymbol{x}(t-1) - \hat{\boldsymbol{\mu}}_{v_{i,j}}(t-1)\right)^H \left(\Sigma_{v_{i,j}}\right)^{-1}\left(\boldsymbol{x}(t) - A\boldsymbol{x}(t-1) - \hat{\boldsymbol{\mu}}_{v_{i,j}}(t-1)\right)\right\}\right)}{\partial\hat{\mu}} \\ &= \left(\Sigma_{v_{i,j}}\right)^{-1}\sum_{i,j}\sum_{t=1}^{T} \gamma_{i,j}(t)\left(\hat{\boldsymbol{x}}_{i,j}^{t}(t) - A\hat{\boldsymbol{x}}_{i,j}^{t-1}(t-1) - \hat{\boldsymbol{\mu}}_{v_{i,j}}(t-1)\right) \end{aligned} \tag{F.8}$$

Since we want to calculate $\hat{\mu}_{\boldsymbol{v}_{i,j}}$ for each i, j, we can rewrite (F.8) as

$$\begin{aligned} \frac{\partial Q(\theta,\hat{\theta})}{\partial \hat{\mu}_{\boldsymbol{v}_{i,j}}} = 0 \Rightarrow \sum_{t=1}^{T} \gamma_{i,j}(t)\left(\hat{\boldsymbol{x}}_{i,j}^{\boldsymbol{t}}(t) - A(t)\hat{\boldsymbol{x}}_{i,j}^{\boldsymbol{t-1}}(t-1) - \hat{\mu}_{\boldsymbol{v}_{i,j}}(t-1)\right) = 0 \\ \hat{\mu}_{\boldsymbol{v}_{i,j}} = \frac{\sum_{t=2}^{T} \gamma_{i,j}(t)\left(\hat{\boldsymbol{x}}_{i,j}^{\boldsymbol{t}}(t) - A(t)\hat{\boldsymbol{x}}_{i,j}^{\boldsymbol{t-1}}(t-1)\right)}{\sum_{t=2}^{T}\gamma_{i,j}(t)} \end{aligned} \tag{F.9}$$

Obviously, (F.9) can be obtained for each mixand j as follows (by combining K filters).

$$\hat{\mu}_{\boldsymbol{v}_j} = \sum_{i=1}^{K} \hat{\mu}_{\boldsymbol{v}_{i,j}} = \frac{\sum_{i=1}^{K} \sum_{t=2}^{T} \gamma_{i,j}(t)\left(\hat{\boldsymbol{x}}_{i,j}^{t}(t) - A(t)\hat{\boldsymbol{x}}_{i,j}^{t\text{-}1}(t-1)\right)}{\sum_{i=1}^{K} \sum_{t=2}^{T} \gamma_{i,j}(t)} \tag{F.10}$$

In the non-parametric approach, our goal is to estimate the time-varying mean (and variance) using a (defined) basis function Ψ (spline basis function in this paper). As a result, the time-varying means $E\{N_{E_j}(t)\}$ and $E\{N_{I_j}(t)\}$ can be estimated as follows (recall that $E\{\mu_{\boldsymbol{v}_j}(t)\}$=[0, $E\{N_{E_j}(t)\}$, $E\{N_{I_j}(t)\}$]).

$$\begin{aligned} E\left\{N_{E_j}(t)\right\} &= \frac{\sum_{i=1}^{K} \gamma_{i,j}(t)\left(\hat{g}_{E_{i,j}}(t) - a_E \hat{g}_{E_{i,j}}(t-1)\right)}{\sum_{i=1}^{K} \gamma_{i,j}(t)} \\ E\left\{N_{I_j}(t)\right\} &= \frac{\sum_{i=1}^{K} \gamma_{i,j}(t)\left(\hat{g}_{I_{i,j}}(t) - a_I \hat{g}_{I_{i,j}}(t-1)\right)}{\sum_{i=1}^{K} \gamma_{i,j}(t)} \end{aligned} \tag{F.11}$$

where g_E and g_I are the second and third components of the state vector $\boldsymbol{x}$, respectively, $\alpha_E = 1\text{-}dt/\tau_E$ and $\alpha_I = 1\text{-}dt/\tau_I$ are inferred from (4.1). In order to model $E\{N_{E_j}(t)\} = \Psi(t) \times \boldsymbol{\omega}_{E_j}$ and $E\{N_{I_j}(t)\} = \Psi(t) \times \boldsymbol{\omega}_{I_j}$, we need to find the weighting vectors, $\boldsymbol{\omega}_{E_j}$ and $\boldsymbol{\omega}_{I_j}$, of the basis function Ψ as follows:

$$\begin{aligned} \boldsymbol{\omega}_{E_j} &= \left(\Psi^H \Psi\right)^{-1} \Psi^H E\{\boldsymbol{N}_{E_j}\} \\ \boldsymbol{\omega}_{I_j} &= \left(\Psi^H \Psi\right)^{-1} \Psi^H E\{\boldsymbol{N}_{I_j}\} \end{aligned} \tag{F.12}$$

where $E\{\boldsymbol{N}_{E_j}\}$ and $E\{\boldsymbol{N}_{I_j}\}$ indicate $E\{N_{E_j}(t)\}$ and $E\{N_{I_j}(t)\}$ over the entire time. The basis function Ψ consists of 50 spline basis functions, as that in [93].

Similarly, the covariance matrix $\hat{\Sigma}_{\boldsymbol{v}_{i,j}}$ can be inferred by taking the derivative of (F.2) with respect to this parameter.

$$\frac{\partial Q(\theta,\hat{\theta})}{\partial \hat{\Sigma}_{v_{i,j}}^{-1}} = \frac{\partial\left(\sum_{i,j}\sum_{t=1}^{T}\gamma_{i,j}(t)E_{p(X|i,j,Y)}\left\{-\frac{1}{2}\log\hat{\Sigma}_{v_{i,j}} + \left(\boldsymbol{x}(t)-A(t)\boldsymbol{x}(t-1)-\hat{\boldsymbol{\mu}}_{v_{i,j}}(t-1)\right)^{H}\left(\hat{\Sigma}_{v_{i,j}}\right)^{-1}\left(\boldsymbol{x}(t)-A(t)\boldsymbol{x}(t-1)-\hat{\boldsymbol{\mu}}_{v_{i,j}}(t-1)\right)\right\}\right)}{\partial\hat{\Sigma}_{v}^{-1}}$$

$$= \frac{1}{2}\sum_{i,j}\sum_{t=1}^{T}\gamma_{i}(t)\left(\hat{\Sigma}_{v_{i,j}} - \left(\hat{\boldsymbol{x}}_{i,j}^{t}(t)-A(t)\hat{\boldsymbol{x}}_{i,j}^{t-1}(t-1)-\hat{\boldsymbol{\mu}}_{v_{i,j}}(t-1)\right)\left(\hat{\boldsymbol{x}}_{i,j}^{t}(t)-A(t)\hat{\boldsymbol{x}}_{i,j}^{t-1}(t-1)-\hat{\boldsymbol{\mu}}_{v_{i,j}}(t-1)\right)^{H}\right) \qquad (F.13)$$

Then, for each *i* and *j* we have

$$\frac{\partial Q(\theta,\hat{\theta})}{\partial \hat{\Sigma}_{v_{i,j}}^{-1}} = \frac{1}{2}\sum_{t=1}^{T}\gamma_{i,j}(t)\left(\hat{\Sigma}_{v_{i,j}} - \left(\hat{\boldsymbol{x}}_{i,j}^{t}(t)-A(t)\hat{\boldsymbol{x}}_{i,j}^{t-1}(t-1)-\hat{\boldsymbol{\mu}}_{v_{i,j}}(t-1)\right)\left(\hat{\boldsymbol{x}}_{i,j}^{t}(t)-A(t)\hat{\boldsymbol{x}}_{i,j}^{t-1}(t-1)-\hat{\boldsymbol{\mu}}_{v_{i,j}}(t-1)\right)^{H}\right) = 0$$

$$\hat{\Sigma}_{v_{i,j}} = \frac{\sum_{t=2}^{T}\gamma_{i,j}(t)\left(\hat{R}_{i,j}^{t}(t) - \left[A(t)\ \ \hat{\boldsymbol{\mu}}_{v_{i,j}}(t-1)\right]\left[\hat{R}_{i,j}^{t-1}(t)\ \ \hat{\boldsymbol{x}}_{i,j}^{t}(t)\right]^{H}\right)}{\sum_{t=2}^{T}\gamma_{n,m}(t)} \qquad (F.14)$$

Note that the full derivation of (F.14) is described in [69]. In a manner similar to (F.9), (F.14) can be obtained for each mixand *j* as follows,

$$\hat{\Sigma}_{v_j} = \frac{\sum_{i=1}^{K}\sum_{t=2}^{T}\gamma_{i,j}(t)\left(\hat{R}_{i,j}^{t}(t) - \left[A(t)\ \ \hat{\boldsymbol{\mu}}_{v_{i,j}}(t-1)\right]\left[\hat{R}_{i,j}^{t-1}(t)\ \ \hat{\boldsymbol{x}}_{i,j}^{t}(t)\right]^{H}\right)}{\sum_{i=1}^{K}\sum_{t=2}^{T}\gamma_{i,j}(t)} \qquad (F.15)$$

Recalling (4.11), the time varying variances of the excitatory and inhibitory inputs can be expressed as

$$\hat{\Sigma}_{E_j}(t)=\frac{\sum_{i=1}^{K}\gamma_{i,j}(t)\left(\hat{R}_{i,j}^{t}(t)\Big|_{2,2}-\left[a_E \quad N_{E_{i,j}}(t)\right]\left[\hat{R}_{i,j}^{t-1}(t)\Big|_{2,2} \quad \hat{g}_{E_{i,j}}(t)\right]^H\right)}{\sum_{i=1}^{K}\gamma_{i,j}(t)}$$
$$\hat{\Sigma}_{I_j}(t)=\frac{\sum_{i=1}^{K}\gamma_{i,j}(t)\left(\hat{R}_{i,j}^{t}(t)\Big|_{3,3}-\left[a_I \quad N_{I_{i,j}}(t)\right]\left[\hat{R}_{i,j}^{t-1}(t)\Big|_{3,3} \quad \hat{g}_{I_{i,j}}(t)\right]^H\right)}{\sum_{i=1}^{K}\gamma_{i,j}(t)} \tag{F.16}$$

where $\hat{R}_{i,j}^{t}(t)\Big|_{2,2}$ stands for the (2, 2)$^{\text{th}}$ element of the matrix $\hat{R}_{i,j}^{t}(t)$. In our non-parametric model, $\hat{\Sigma}_{E_j}(t)=\Psi(t)\,\lambda_{E_j}$ and $\hat{\Sigma}_{I_j}(t)=\Psi(t)\,\lambda_{I_j}$. The weighting vectors λ_{E_j} and λ_{E_j} of the basis function Ψ are given by

$$\begin{aligned}\boldsymbol{\lambda}_{E_j}&=\left(\Psi^H\Psi\right)^{-1}\Psi^H\hat{\boldsymbol{\Sigma}}_{E_j}\\ \boldsymbol{\lambda}_{I_j}&=\left(\Psi^H\Psi\right)^{-1}\Psi^H\hat{\boldsymbol{\Sigma}}_{I_j}\end{aligned} \tag{F.17}$$

where $\hat{\boldsymbol{\Sigma}}_{E_j}$ and $\hat{\boldsymbol{\Sigma}}_{I_j}$ indicate $\hat{\Sigma}_{E_j}(t)$ and $\hat{\Sigma}_{I_j}(t)$ over the entire time, respectively. Hence, the system noise (including synaptic inputs) can be represented by a Gaussian mixture model (GMM) as follows.

$$\boldsymbol{v}(t)\approx\sum_{j=1}^{G}\hat{\alpha}_j N\left(\hat{\boldsymbol{\mu}}_{v_j}(t),\hat{\Sigma}_{v_j}(t)\right) \tag{F.18}$$

Moreover, as mentioned before, we can simply calculate the final state estimate $\boldsymbol{x}(t)$ as the combination of $K\times G$ parallel filters at each time t.

$$\hat{\boldsymbol{x}}(t)=\sum_{i,j}\gamma_{i,j}(t)\hat{\boldsymbol{x}}_{i,j}^{t}(t) \tag{F.19}$$

Note that the variances of the membrane voltage (σ^2_w) and observation noise (σ^2_ε) can be calculated in a straightforward way, as mentioned in [93].